

ACADEMIA DE STUDII ECONOMICE

Facultatea: Cibernetică, Statistică și Informatică Economică

Specializarea: **Cibernetică Economică**



Sisteme Suport de decizie:

PROIECT SEMINAR:

**Aplicarea Sistemului Suport de Decizie în cadrul organizației
“Căile Ferate Române”**

Cărtuță Rareș-Constantin

Grupa 1075, Anul III

Profesor Coordonator:

Georgescu Irina-Alexandra

Cuprins

Introducere:	3
Partea 1 - Managementul bazelor de date	3
1.1 Conceperea bazei de date	3
1.2 Curățarea datelor și popularea bazei de date	8
1.2.1. Sucursale și Funcții:	8
1.2.2 Angajați	9
1.2.3 Achiziții și Furnizori	12
1.3 Analiză descriptivă scurtă a datelor în R	17
Partea 2 - Modelarea și analiza datelor	19
2.1 Analiza cu Charts și Dashboard în Power BI	19
2.2 Prognoza unor indicatori micro- sau macro economici	21
2.3 Rezolvarea unei probleme decizionale economice în cadrul respectivei companii	24
Partea 3 - Interfața cu utilizatorul	28
Partea 4- Managementul cunoștințelor	32
Concluziile analizei	33
Surse web pentru date	33

Introducere:

În cadrul acestui proiect, am urmărit analiza situației financiare a organizației “Căile Ferate Române”, pe baza datelor publice de pe site-ul CFR Călători privind achiziții publice, date salariale și date de natură financiară.

Am stocat toate informațiile pe care le-am considerat relevante în realizarea aplicației într-o bază de date relațională de tip SQL, prin intermediul aplicației “Microsoft SQL Server Management Studio”.

Pentru curățarea și standardizarea datelor de interes, cât și efectuarea unei scurte analize descriptive a datelor, am folosit aplicațiile EXCEL și RSTUDIO.

Pentru vizualizarea datelor într-un format tip dashboard, am folosit aplicația Power BI.

De asemenea, am folosit Excel în scopul realizării unei previziuni asupra unui indicator economic, dar și în cadrul unei probleme decizionale care necesita o scurtă analiză de text, urmată de aplicare metodei Lanțurilor Markov, în această analiză de text, a fost folosită, în prima fază, aplicația ANT CONC.

Partea 1 - Managementul bazelor de date

1.1 Conceperea bazei de date

În cadrul proiectului, mi-am propus să extrag și să standardizez cele mai relevante informații legate de funcționarea firmei. În acest scop, am creat 5 tabele, cu legături între fiecare dintre ele (Sucursale, Funcții, Angajați, Furnizori, Achiziții).

Datele extrase sunt publice, fiind luate de pe site-ul CFR Călători și cuprind:

- Anexele de achiziții publice
- Raportări Contabile
- Bugetul Salarial

Descrierea Tabelelor:

Tabela Sucursală conține două câmpuri: id_sucursală și denumire_sucursală, în cadrul acesteia sunt incluse cele 8 sucursale CFR (București, Craiova, Timișoara, Cluj, Brașov, Iași, Galați, Constanța). Tabelul are o legătură indirectă de tip one to many cu tabelul Achiziții și încă una cu tabelul Angajați.

```

Create Table Sucursale
(
    id_sucursala int identity(1,1) primary key,
    denumire_sucursala varchar(250)
);

```

Fig 1.1 Query creare pentru tabela Sucursală

id_sucursala	denumire_sucursala
1	Bucuresti
2	Craiova
3	Timisoara
4	Cluj
5	Brasov
6	Iasi
7	Galati
8	Constanta

Fig 1.2 Conținut Tabelă Sucursală

Tabela Funcții a fost creată pe baza informațiilor salariale aflate pe site-ul CFR Călători. Aceasta conține 4 câmpuri: id_funcție, denumire_funcție, salariu_baza_brut_min, salariu_baza_brut_max și tip_funcție. Cele 2 câmpuri reprezintă limitele în care se poate încadra salariul de baza brut al angajaților care ocupă o anumită funcție, iar câmpul tip_funcție reprezintă o variabilă categorială care poate facilita vizualizarea și prelucrarea informațiilor. Tabela are o legătură indirectă de tip one to many cu Angajați. Conține 95 de funcții

```

Create Table Functii
(
    id_funcție int identity(1,1) primary key,
    denumire_funcție varchar(250),
    salariu_baza_brut_min int,
    salariu_baza_brut_max int,
    tip_funcție varchar(250)
);

```

Fig 1.3 Query pentru crearea Tabelei Funcții

id_funcție	denumire_funcție	salariu_baza_brut_min	salariu_baza_brut_max	tip_funcție
1	Muncitor necalificat	3293	3376	muncitor_necalificat
2	Muncitor calificat grad I	3783	4308	muncitor_calificat
3	Muncitor calificat grad II	3455	3855	muncitor_calificat
4	Vânzător bilete	3455	3520	activitate_exploatare
5	Acar	3520	3701	activitate_exploatare
6	Conducător tren	3666	3701	activitate_exploatare
7	Informator călători	3455	3520	activitate_exploatare
8	Magaziner comercial	3520	3590	activitate_exploatare
9	Manevrant vagoane	3666	3855	activitate_exploatare
10	Șef tură la comanda personalului de tren	4532	4630	activitate_exploatare
11	Șef manevră	3943	4120	activitate_exploatare

Fig 1.4 Conținut tabelă Funcții

Tabela Angajați este oarecum specială, deoarece datele din interiorul acesteia sunt practic doar o simulare, acestea fiind generate în urma rulării unui script în R, procesul populării acestui tabel va fi descris în subcapitolele următoare.

Tabela conține 9 coloane dintre care id_angajat, gen, nume, prenume, email, telefon, salariu, id_sucursala, id_functie.

Gen conține un constraint care permite doar valorile “F” și “M”, pentru facilitarea simulării datelor, Telefon conține un constraint care permite doar introducerea cifrelor, iar id_sucursala și id_functie reprezintă foreign key-uri prin intermediul cărora se realizează legătura directă între Angajați și tabela Sucursale, respectiv tabela Funcții.

Tabela conține 2000 de intrări.

```

]Create Table Angajati
(
    id_angajat int identity(1,1) primary key,
    gender VARCHAR(250) CHECK (gender IN ('M', 'F')),
    nume varchar(250),
    prenume varchar(250),
    email varchar(250),
    telefon CHAR(10),
    CONSTRAINT chk_telefon CHECK (telefon not like '%[^0-9]%'),
    id_sucursala int,
    CONSTRAINT FK_angajati_sucursale FOREIGN KEY (id_sucursala)
    REFERENCES Sucursale (id_sucursala),
    id_functie int,
    CONSTRAINT FK_angajati_functii FOREIGN KEY (id_functie)
    REFERENCES Functii (id_functie),
    salariu int
);

```

Fig 1.5: Query Pentru crearea tabeli Angajați

id_angajat	gender	nume	prenume	email	telefon	id_sucursala	id_functie	salariu
1	M	CROITORU	Giorgian	Giorgian.CROITORU@yahoo.com	0785545509	3	12	2755
3	M	CRACIUN	Ducu	Ducu.CRACIUN@yahoo.com	0744253946	5	22	3145
4	M	NITA	Cristinel	Cristinel.NITA@yahoo.com	0785115459	7	60	4679
5	M	IORGA	Zaharia	Zaharia.IORGA@yahoo.com	0799692898	3	65	5904
6	F	NICULESCU	Anghelina	Anghelina.NICULESCU@yahoo.com	0779722234	4	81	5810
7	M	IACOB	Leordean	Leordean.IACOB@yahoo.com	0797817892	4	13	2632
8	M	GROZA	Igor	Igor.GROZA@yahoo.com	0783523669	1	73	3769
9	F	NISTOR	Andrada	Andrada.NISTOR@yahoo.com	0733101049	1	43	5168
10	F	PETRESCU	Camelia	Camelia.PETRESCU@yahoo.com	0763772506	1	36	2252
11	F	COJOCARU	Marioara	Marioara.COJOCARU@yahoo.com	0756083283	4	63	5664
13	M	NEAGU	Ianis	Ianis.NEAGU@yahoo.com	0764294367	8	34	1993
14	F	IGNAT	Marcheta	Marcheta.IGNAT@yahoo.com	0723884264	7	69	3635
15	M	FILIP	Titus	Titus.FILIP@gmail.com	0793845868	1	83	6434
16	M	UNGUREANU	Florin	Florin.UNGUREANU@yahoo.com	0743629762	1	67	6031
17	F	ADAM	Cleopatra	Cleopatra.ADAM@yahoo.com	0771908601	6	74	4065
18	M	MIRON	Visarion	Visarion.MIRON@gmail.com	0742912986	7	3	2301
20	F	VERES	Flora	Flora.VERES@gmail.com	0713176863	7	9	2359
21	F	PATRASCU	Tamara	Tamara.PATRASCU@yahoo.com	0703141082	1	52	4366
23	M	NICULESCU	Marinel	Marinel.NICULESCU@yahoo.com	0741619993	4	20	2988

Fig 1.6: Conținut Tabela Angajați

Tabela Furnizori are doar două câmpuri(id_furnizor și denumire_furnizor) și a fost creată pe baza datelor din anexele de achiziții, această există pentru a facilita standardizarea și curățarea datelor, lucru care va fi evidențiat în subcapitolele următoare.

În urma standardizării datelor tabela are 1047 de înregistrări.

```
Create Table Furnizori
(
    id_furnizor int identity(1,1) primary key,
    denumire_furnizor varchar(250),
)
```

Fig 1.7: Query creare tabelă Furnizori

id_furnizor	denumire_furnizor
1	SC APA TALEA SRL
2	SC VIOSIL INSTALCONSTRUCT SRL
3	SC BRAMCO CONRECON SRL
4	SC SCUDIVER TOOLS SRL
5	SC RIAN CONSULT SRL
6	SC EMON SRL
7	SPITALUL CLINIC CF 2 BUCURESTI
8	SC ROMOLD SRL
9	SC ALSIMCRIS UTIL SRL
10	SC TISECO SRL
11	SC EURO VIAL LIGHTING SRL
12	SC PSICOM GLOBAL SRL
13	SC PIN PLUS PIN SRL
14	SC SERVICII SALUBRITATE BUCURESTI SA
15	SC GRAFORESS SRL
16	SC OZON SERV SRL
17	SC POLYGON TRADING SRL
18	SC SAFETY SECURITY SRL
19	SC MASTERY ELECTRONICS SRL

Fig 1.8: Conținut tabelă Furnizori

Tabela Achiziții este cea mai vastă din baza de date, iar standardizarea și curățarea datelor din ea a ocupat cea mai mare parte din proiect. Aceasta cuprinde o bună parte din totalul achizițiilor publice din cele 8 sucursale din 2015 până în prezent, cuprinzând 2331 de intrări. Tabela conține informații cu privire la data de începere și finalizare a unei achiziții, obiectivul său, tipul său, valoarea sa, stadiul de execuție și durata în zile.

De asemenea, tabela conține două foreign key-uri: id_furnizor și id_sucursală, prin intermediul cărora se face legătură cu tabela Furnizori, respectivă Sucursale. Ambele chei redau o relație de tip one to many.

```

)Create Table Achizitii
(
    id_achizitie int identity(1,1) primary key,
    id_sucursala int,
    constraint FK_achizitii_sucursale foreign key (id_sucursala)
    references Sucursale (id_sucursala),
    data_achizitie date,
    data_finalizare date,
    obiect varchar(MAX),
    valoare float,
    durata_zile int,
    tip_procedura varchar(MAX),
    id_furnizor int,
    constraint FK_achizitii_furnizori foreign key (id_furnizor)
    references Furnizori (id_furnizor),
    stadiu_executie DECIMAL(5, 2)
)

```

Figura 1.9: Query creare tabelă Achiziții

id_achizitie	id_sucursala	data_achizitie	data_finalizare	obiect	valoare	durata_zile	tip_procedura	id_furnizor	stadiu_executie
1	1	2016-01-04	2017-01-03	Furnizare apa pentru dozator Post Revizie Vag. Ploiest...	5036.64	365	Achizitie directa	1	1.00
2	1	2016-01-21	2016-06-20	Cartuse toner pentru imprimante, copiatoare si aparate ...	42473	151	Cerere de oferta	697	1.00
3	1	2016-01-27	2016-03-26	Executie instalatii incalzire a cladirilor din grupa tehnica...	119576.39	59	Licitatie deschisa	2	1.00
4	1	2016-01-28	2016-04-27	Realizarea retelei de canalizare a apelor pluviale cu m...	45320.2	90	Cerere de oferta	3	1.00
5	1	2016-02-01	2016-06-28	Lenjerie de pat, cearceaf plic, cearceaf pat, fata de pe...	274200	148	Cerere de oferta	398	1.00
7	1	2016-03-08	2016-04-07	Reparare a doua vinciuri (20tf) la Revizia de vag. Buc...	15032.8	30	Achizitie directa	4	1.00
8	1	2016-03-10	2017-03-09	Servicii de colectare si eliminare filtre si lavete uzate la ...	3792	364	Achizitie directa	5	1.00
9	1	2016-03-17	2016-04-16	Repararea statiei IT la Revizia de vag. Bucuresti Basa...	17505	30	Achizitie directa	6	1.00
10	1	2016-04-14	2016-07-13	Acumulatori 12V pentru sursa neinteruptibila de putere...	1222	90	Licitatie deschisa	668	1.00
11	1	2016-04-14	2017-05-08	Prestarea serviciilor de examinari medicale si psihologi...	19266	389	Licitatie deschisa	56	1.00
12	1	2016-04-14	2017-05-08	Prestarea serviciilor de examinari medicale si psihologi...	104305.5	389	Licitatie deschisa	7	1.00
13	1	2016-04-19	2016-06-18	Serviciul de paza a obiectivului, bunurilor si valorilor di...	86572.8	60	Negociere	8	1.00
14	1	2016-04-19	2016-05-08	Executarea lucrarilor de reparare ferestre termopan la s...	6131	19	Achizitie directa	9	1.00
15	1	2016-04-19	2016-06-03	Lucrari de curatare si calibrare rezervoare motorina di...	70988.4	45	Achizitie directa	10	1.00
16	1	2016-04-25	2016-05-31	Furnizare de megohmetru 5000V la Depoul Ploiesti	8762.9	36	Achizitie directa	11	1.00
20	1	2016-05-23	2016-06-22	Echipament tip CUTTER PLOTTER profesional la Re...	20513.37	30	Achizitie directa	13	1.00
21	1	2016-05-24	2016-11-23	Costume salopeta - 1350 buc	63990	183	Cerere de oferta	213	1.00
22	1	2016-05-30	2016-06-29	Ridicare transport si depozitare deseuri menajere si inc...	3907.6	30	Negociere	14	1.00
23	1	2016-06-07	2017-06-14	Buletine de avizare a restrictiilor de viteza	38491.2	372	Licitatie deschisa	15	1.00

Figura 1.10: Conținut tabelă Achiziții

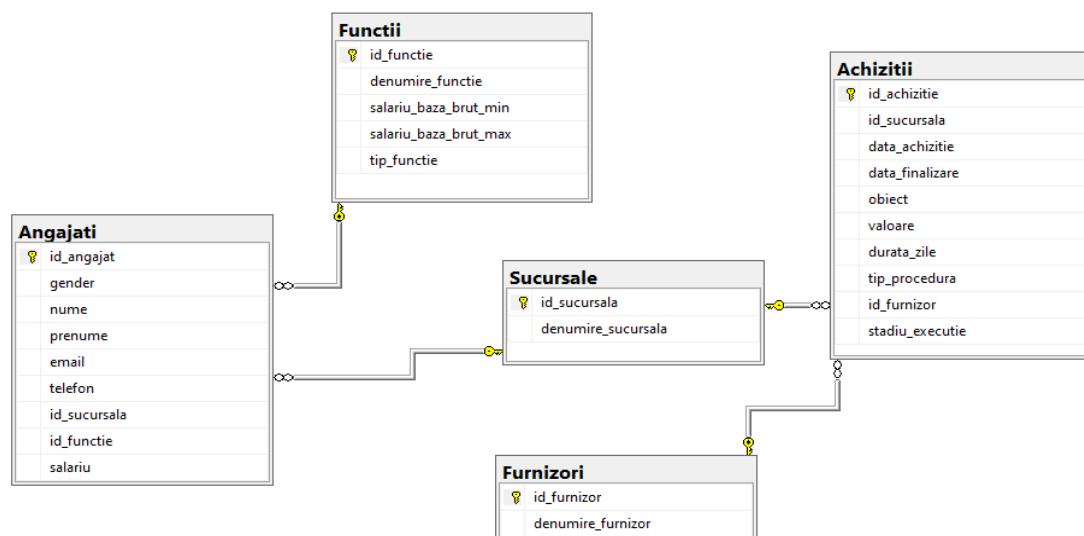


Figura 1.11: Schema Conceptuală a bazei de date

1.2 Curățarea datelor și popularea bazei de date

Curățarea și standardizarea datelor reprezintă o etapă esențială în efectuarea oricărei analize a datelor și care poate să ocupe enorm de mult timp. În cadrul acestei etape, mi-am propus să păstrez cât mai multe date cu putință, așadar a fost nevoie să implementez o oarecare automatizare în cadrul populării, dar și al curățării datelor.

Modul în care am realizat aceasta etapă a proiectului, diferă de la tabelă, la tabelă, spre exemplu sucursale și funcții au putut fi introduse manual, fără prea mari probleme, în schimb pentru tabela angajați a fost nevoie de multă improvizație, datele din acestea fiind generate aleator pe baza unui script, iar pentru curățarea tabelii Achiziții a fost nevoie de supunerea la o curățare inițială în Excel, urmată de implementarea a numeroase funcții în R, dar și a uneia de SQL, pe care am invocat-o ulterior în R.

Detalierea Procesului de Populare:

1.2.1. Sucursale și Funcții:

Cele două tabele au fost populate în cea mai facilă manieră, fiind necesare doar câteva interogări de tipul INSERT.

```

INSERT INTO Sucursale(denumire_sucursala)
VALUES('Bucuresti')
INSERT INTO Sucursale(denumire_sucursala)
VALUES('Craiova')
INSERT INTO Sucursale(denumire_sucursala)
VALUES('Timisoara')
INSERT INTO Sucursale(denumire_sucursala)
VALUES('Cluj')
INSERT INTO Sucursale(denumire_sucursala)
VALUES('Brasov')
INSERT INTO Sucursale(denumire_sucursala)
VALUES('Iasi')
INSERT INTO Sucursale(denumire_sucursala)
VALUES('Galati')
INSERT INTO Sucursale(denumire_sucursala)
VALUES('Constanta')

```

Figura 1.12: Populare Sucursale

```

INSERT INTO Functii (denumire_functie, salariu_baza_brut_min, salariu_baza_brut_max, tip_functie)
VALUES
('Muncitor necalificat', 3293, 3376, 'muncitor_necalificat'),
('Muncitor calificat grad I', 3783, 4308, 'muncitor_calificat'),
('Muncitor calificat grad II', 3455, 3855, 'muncitor_calificat'),
('Vânzător bilete', 3455, 3520, 'activitate_exploatare'),
('Acar', 3520, 3701, 'activitate_exploatare'),
('Conducător tren', 3666, 3701, 'activitate_exploatare'),
('Informator călători', 3455, 3520, 'activitate_exploatare'),
('Magaziner comercial', 3520, 3590, 'activitate_exploatare'),
('Manevrant vagoane', 3666, 3855, 'activitate_exploatare'),
('Sef tură la comanda personalului de tren', 4532, 4630, 'activitate_exploatare'),
('Sef manevră', 3943, 4120, 'activitate_exploatare'),

```

Figura 1.13: Populare Funcții

1.2.2 Angajați

În lipsa accesului publicului la datele angajaților, am recurs la simularea acestora, pe baza informațiilor extrase în tabela Funcții, cu ajutorul mediului de programare RSTUDIO.

Pentru început am realizat 3 documente text, în care am inclus cele mai comune prenume (masculine, cât și feminine, fiecare într-un folder separat) și nume românești, apoi le-am importat în R.

```
> length(prenume_barbati)
[1] 329
> length(prenume_barbati)
[1] 329
> length(prenume_femei)
[1] 439
> length(ume_romanesti)
[1] 300
> prenume_barbati[1:10]
[1] "AbeI"    "Achim"  "Adam"   "Adelin" "Adi"    "Adonis" "Adrian" "Agnos"  "Albert"
[10] "Aleodor"
> prenume_femei[1:10]
[1] "Ada"    "Adela"  "Adelaida" "Adelina" "Adina"  "Adriana" "Agata"  "Aglaiia"
[9] "Agripina" "Aida"
> ume_romanesti[1:10]
[1] "POP"    "POPESCU" "POP"    "RADU"    "IONESCU" "DUMITRU" "STOICA" "STAN"
[9] "GHEORGHE" "RUSU"
> |
```

Figura 1.14: Nume Introduse în R

Apoi am folosit funcția `sample()` din R pentru a genera un set de “genuri”

```
> dom <- sample(1:2, size=500, replace=TRUE, prob=c(0.5,0.5))
> genuri <- factor(dom, labels=c("F","M"))
> genuri[1:10]
[1] M M M F F F M M F F
Levels: F M
> length(genuri)
[1] 500
```

Figura 1.15: Generare genuri

După obținerea genurilor acestea pot fi folosite pentru a condiționa generarea prenumelor în funcție de genul persoanei.

```
> n_elements <- length(ume_romanesti)
> equal_probabilities <- rep(1/n_elements, n_elements)
> dom <- sample(1:n_elements, size=500, replace=TRUE, prob=equal_probabilities)
> nume_generate <- ume_romanesti[dom]
> prenume_generate <- ifelse(genuri == "F", sample(prenume_femei, size = 500, replace = TRUE), sample(prenume_barbati, size = 500, replace = TRUE))
> prenume_generate[1:10]
[1] "Grigore" "Brăduț" "Dacian" "Fabia" "Alexia" "Corina" "Petrișor" "Iorgu"
[9] "Ștefana" "Anișoara"
> nume_generate[1:10]
[1] "TOADER" "STANESCU" "PETRE" "NICOLAE" "RUSU" "BUNEA" "SIMA" "PIRVU"
[9] "BUNEA" "ILIESCU"
```

Figura 1.16: Generare prenume și nume

Am continuat prin generarea adreselor de email, dând o probabilitate de 80% ca adresa angajatului să fie de la yahoo și o probabilitate de 20% ca aceasta să fie de gmail.

```
> email_domains <- sample(c("@gmail.com", "@yahoo.com"), size = 500, replace = TRUE, prob = c(0.2, 0.8))
> emails_generate <- paste0(prenume_generate, ".", nume_generate, email_domains)
> emails_generate[1:10]
[1] "Grigore.TOADER@yahoo.com" "Brăduț.STANESCU@yahoo.com" "Dacian.PETRE@yahoo.com"
[4] "Fabiă.NICOLAE@yahoo.com" "Alexia.RUSU@yahoo.com" "Corina.BUNEA@yahoo.com"
[7] "Petrișor.SIMA@yahoo.com" "Iorgu.PIRVU@yahoo.com" "Ștefana.BUNEA@yahoo.com"
[10] "Anișoara.ILIESCU@yahoo.com"
```

Figura 1.17: Generare email

În continuare sunt generate numerele de telefon, acestea încep de fiecare dată cu 07, iar celelalte 8 cifre sunt generate aleator.

```
> numere_telefon_generate <- character(2000)
>
> for (i in 1:2000) {
+   random_digits <- sample(0:9, size = 8, replace = TRUE)
+   numere_telefon <- paste0("07", paste0(random_digits, collapse = ""))
+   numere_telefon_generate[i] <- numere_telefon
+ }
> numere_telefon_generate[1:10]
[1] "0749496990" "0763066800" "0764522210" "0797946135" "0709134294" "0757329949" "0751823946"
[8] "0710271426" "0784309834" "0757911043"
> |
```

Figura 1.18: Generare Telefon

Pentru generarea id-urilor de tip funcție, este necesară comunicarea cu baza de date, acest lucru este posibil folosind librăria RODBC.

```
> dbhandle <- odbcDriverConnect('driver={SQL Server};server=██████████;database=Proiect_SSd;trusted_connection=true')
> query_functii <- sqlQuery(dbhandle, 'select * from Functii')
> query_sucursale <- sqlQuery(dbhandle, 'select * from Sucursale')
> odbcClose(dbhandle)
> dom <- sample(1:8, size=2000, replace=TRUE, prob=c(0.2,0.1,0.1,0.1,0.1,0.1,0.1,0.1))
> sucursale <- query_sucursale$id_sucursala
> sucursale_generate <- factor(sucursale[dom], levels = sucursale)
> table(sucursale_generate)
sucursale_generate
 1  2  3  4  5  6  7  8
445 233 230 220 205 213 213 241
```

Figura 1.19: Sucursale generate în urma interogării

```
> n_elements <- sum(table(query_functii$id_functie))
> equal_probabilities <- rep(1/n_elements, n_elements)
> dom <- sample(1:n_elements, size = 2000, replace = TRUE, prob = equal_probabilities)
> functii <- query_functii$id_functie
> functii_generate <- factor(dom, levels = 1:n_elements, labels = functii)
> functii_generate[1:10]
[1] 69 81 36 78 17 28 32 2 69 81
```

Figura 1.20: Funcții generate în urma interogării

Pentru generarea salariului este nevoie de un pic mai multă muncă, componenta aleatorie are loc de fapt în cadrul unei interogări SQL, unde se citește id-ul funcției angajatului apoi se returnează suma dintre salariul minim al funcției respective și produsul dintre un număr aleatoriu cu valori între 0 și 1 și diferența dintre salariul maxim și cel minim, apoi, odată ce

acest număr este returnat, aceasta urmează să fie transformat în salariul real, în urma aplicării formulei de calculare salarială.

```
> n_elements <- length(funcții_generate)
> salariu_mediu_values <- numeric(n_elements)
> dbhandle <- odbcDriverConnect('driver={SQL Server};server=██████████;database=Proiect_SSD;trusted_connection=true')
> for (i in 1:n_elements) {
+   sql_query <- paste("SELECT ROUND(salariu_baza_brut_min + ((salariu_baza_brut_max - salariu_baza_brut_min) * RAND()), 0) as salariu_brut",
+                     "FROM funcții",
+                     "WHERE id_funcție =", funcții_generate[i])
+   # Execute SQL query
+   result <- sqlQuery(dbhandle, sql_query)
+   avg_salary <- result
+   salariu_mediu_values[i] <- round(avg_salary - (35 * avg_salary / 100) - (10 * (avg_salary - (35 * avg_salary / 100)) / 100), 0)
+ }
> odbcClose(dbhandle)
> salariu_mediu_values[1:10]
[[1]]
[1] 3546

[[2]]
[1] 5316

[[3]]
[1] 2176
```

Figura 1.21: Salarii reale generate

După aceea, am creat un dataframe, cu toate aceste date și pe baza lui am generat un fișier csv.

```
date_angajati <- data.frame(
  id = id,
  nume_generate = nume_generate,
  prenume_generate = prenume_generate,
  genuri = genuri,
  emails_generate = emails_generate,
  numere_telefon_generate = numere_telefon_generate,
  sucursale_generate = sucursale_generate,
  funcții_generate = funcții_generate
)

write.csv(date_angajati, "date_angajati.csv", row.names = FALSE)
```

Figura 1.22: Creare dataframe + CSV

d	nume_generate	prenume_generate	genuri	emails_generate	numere_telefon	sucursale	funcții_ge	salariu_mediu_values
1	CROITORU	Giorgian	M	Giorgian.CROITORU@yahoo.com	785545509	3	12	2755
2	GRIGORE	Brăduț	M	Brăduț.GRIGORE@yahoo.com	754029424	8	32	2351
3	CRACIUN	Ducu	M	Ducu.CRACIUN@yahoo.com	744253946	5	22	3145
4	NITA	Cristinel	M	Cristinel.NITA@yahoo.com	785115459	7	60	4679
5	IORGA	Zaharia	M	Zaharia.IORGA@yahoo.com	799692898	3	65	5904
6	NICULESCU	Anghelina	F	Anghelina.NICULESCU@yahoo.com	779722234	4	81	5810
7	IACOB	Leordean	M	Leordean.IACOB@yahoo.com	797817892	4	13	2632
8	GROZA	Igor	M	Igor.GROZA@yahoo.com	783523669	1	73	3769
9	NISTOR	Andrada	F	Andrada.NISTOR@yahoo.com	733101049	1	43	5168
10	PETRESCU	Camelia	F	Camelia.PETRESCU@yahoo.com	763772506	1	36	2252
11	COJOCARU	Marioara	F	Marioara.COJOCARU@yahoo.com	756083283	4	63	5664
12	MUNTEANU	Brăduț	M	Brăduț.MUNTEANU@yahoo.com	708131289	8	45	4020

Figura 1.23: CSV Angajați

Apoi, am automatizat procesul de inserare a valorilor în baza de date, folosind următorul cod:

```
dbhandle <- odbcDriverConnect('driver={SQL Server};server= database=Proiect_SSD;trusted_connection=true')
for (i in 1:nrow(date_angajati)) {
  query <- paste("INSERT INTO Angajati (gender, nume, prenume, email, telefon, id_sucursala, id_functie, salariu) VALUES (",
    "", date_angajati$genuri[i], "",
    "", date_angajati$nume_generate[i], "",
    "", date_angajati$prenume_generate[i], "",
    "", date_angajati$emails_generate[i], "",
    "", date_angajati$numere_telefon_generate[i], "",
    date_angajati$sucursale_generate[i], "",
    date_angajati$functii_generate[i], "",
    date_angajati$salariu_mediu_values[i], ")",
    sep="")
  odbcquery(dbhandle, query)
}
odbcClose(dbhandle)
```

Figura 1.24: Cod pentru automatizarea inserării

1.2.3 Achiziții și Furnizori

Standardizarea acestor date a reprezentat cea mai dificilă parte a întregului proiect, deoarece datele originale erau pline de inconsistențe și câteodată chiar de greșeli logice.

Nr. Crt.		Nr. și data	Obiect	Valoare (lei, fără TVA)	Durată	Tipul procedurii aplicate pt. atribuire	Denumire furnizor / prestator / executant	Stadiu execuție (%)	Obs
1	1/04.01.2016		Furnizare apa pentru dozator Post Revizie Vag. Ploiesti - 40 recipienti bidon 19L/luna; inchiriere dozator la Post Revizie Vag. Ploiesti - 1 buc., igienizare dozator	5.036,64	12 luni 04.01.2016 - 0.3.01.2017	achizitie directa	SC APA TALEA SRL	100%	
2	3/21.01.2016		Cartuse toner pentru imprimante, copiatoare si aparate fax	42.473,00	6 luni 21.01.2016 - 20.06.2016	cerere de oferte	SC ALIMAR OEM SRL	100%	
3	4/27.01.2016		Executie instalatii incalzire a cladirilor din grupa tehnica Revizia de Vagoane Bucuresti Basarab	119.576,39	60 zile 27.01.2016 - 26.03.2016	licitatie deschisa	SC VIOSIL INSTALCONSTRUCT SRL	100%	
4	5/28.01.2016		Realizarea retelei de canalizare a apelor pluviale cu montarea unui separator - Dep. BC, SELC Basarab	45.320,20	90 zile 28.01.2016 - 27.04.2016	cerere de oferte	SC BRAMCO CONRECON SRL	100%	
5	6/01.02.2016		Lenjerie de pat, cearceaf plic, cearceaf pat, fata de perna	274.200,00	4 luni 29.02.2016 - 28.06.2016	cerere de oferte	SC ELITEX SRL	100%	
6	7/17.02.2016		Lucrari de instalare de echipamente de incalzire centrala, instalatii electrice aferente si instalatii de utilizare de gaze naturale la : cladirea anexa salubritate parter+etaj, remiza PSI+centrul de calcul, cladire baterii acumulatori, cladire magazie materiale+cladire compresor, district IT(dispecerat Mecanic si electric hala), lucrari neefectuate -Revizia de Vagoane Bucuresti Grivita inclusiv obtinere de Autorizatie de Construire	259.179,02	8 luni 17.02.2016 - 16.02.2016	licitatie deschisa	SC AB INSTAL SRL	100%	
7	9/08.03.2016		Reparare a doua vinciuri (20tf) la Revizia de vag. Bucuresti Basarab	15.032,80	30 zile 08.03.2016 - 07.04.2016	achizitie directa	SC SCUDIVER TOOLS SRL	100%	

Figura 1.25: Format Inițial al unei achiziții

Pentru început am descărcat toate datele disponibile și le-am introdus în Excel, apoi am creat o nouă coloana pentru a trece data de începere/data achiziției, folosind următoarea formulă:

“=RIGHT(B6, LEN(B6) - FIND("/", B6))”

Obținerea datei finale s-a dovedit mai dificilă, deoarece aceasta nu era menționată de multe ori, în coloana “Nr și data” sau “Durată” prin urmare am improvisat formula:

“=IFERROR(RIGHT(E6, LEN(E6) - FIND("-", E6)), E6)”

Am creat, de asemenea și o coloană în care este precizată sucursala care a efectuat achiziția.

Acesta este rezultatul final:

Nr. Crt.	Nr. și data	Obiect	Valoare (lei, fără TVA)	Durată	Tipul procedurii aplicate pt. atribuire	Denumire furnizor / prestator executant	Stadiu execuție (%)	Data Începere	Data finalizare/Timp necesar	Sucursala
1	1/04.01.2016	Furnizare apa pentru dozator Post Revizie Vag. Ploiesti - 40 recipienti bidoi 19L luma; inchiriere dozator la Post Revizie Vag. Ploiesti - 1 buc., igienizare dozator	5.036,64	12 luni 04.01.2016 - 03.01.2017	achiziție directă	SC APA TALEA SRL	100%	04.01.2016	03.01.2017	Bucuresti
2	3/21.01.2016	Cartușe toner pentru imprimante, copiatoare și aparate fax	42.473,00	6 luni 21.01.2016 - 20.06.2016	cerere de oferte	SC ALIMAR OEM SRL	100%	21.01.2016	20.06.2016	Bucuresti
3	4/27.01.2016	Execuție instalat în caldura a clădirilor din grupa tehnica Revizia de Vagoane Bucuresti Basarab	119.576,39	60 zile 27.01.2016 - 26.03.2016	licitație deschisă	SC VIOSIL INSTALCONSTRUCT SRL	100%	27.01.2016	26.03.2016	Bucuresti
4	5/28.01.2016	Realizarea rețelei de canalizare a apelor pluviale cu montarea unui separator - Dep. BC, SELC Basarab	45.320,20	90 zile 28.01.2016 - 27.04.2016	cerere de oferte	SC BRAMCO CONRECON SRL	100%	28.01.2016	27.04.2016	Bucuresti
5	6/01.02.2016	Lenjete de pat, cearceaf plic, cearceaf pat, fata de perna	274.200,00	4 luni 01.02.2016 - 28.06.2016	cerere de oferte	SC ELITEX SRL	100%	01.02.2016	28.06.2016	Bucuresti

Figura 1.26: Tabel final date achiziții

Ulterior am creat un alt fișier în care am dat paste as values pentru a nu avea limitări legate de formule. Trebuie să precizez ca acele 2 formule nu au fost suficiente, fiind nevoie să apelez și la SUBSTITUTE() sau la alte variații de LEFT() sau RIGHT(), însă nu mai am acces la ele.

În cadrul acestei capitole nu voi recurge atât de des la print screen-uri cu output ca în cel precedent, deoarece schimbările sunt adesea greu de observat, cea mai stabilă metrică a succesului fiind numărul final de înregistrări din tabelă.

```
date_achizitii <- read_excel("Anexa_Achizitii3.xlsx")
```

```
validare_stadiu_executie <- function(stadiu_executie) {
  stadiu_executie <- ifelse(grepl("%", stadiu_executie), as.numeric(gsub("%", "",
    stadiu_executie)), as.numeric(stadiu_executie))
```

```
  if (is.na(stadiu_executie)) {
    print("format invalid")
    return(NULL)
  }
```

```
  return(ifelse(stadiu_executie > 1, stadiu_executie / 100, stadiu_executie))
}
```

```
date_achizitii$`Stadiu execuție (%)` <- sapply(date_achizitii$`Stadiu execuție (%)`,
  validare_stadiu_executie)
```

În cadrul codului de mai sus a loc citirea fișierului urmată de implementarea unei funcții care convertește valorile în numere cuprinse între 0 și 1. Mai întâi caută caracterul % în cadrul

argumentului, iar apoi elimină caracterul și convertește argumentul într-o valoare numerică, dacă acea valoare numerică este mai mare decât 1, realizează împărțirea la 100.

```
curatare_numere <- function(string_numeric) {
  if (grepl(",", string_numeric)) {
    cleaned_string <- gsub("\\.", "", string_numeric)
    cleaned_string <- gsub(",", ".", cleaned_string)

    cleaned_numeric <- as.numeric(cleaned_string)

    if (!is.na(cleaned_numeric)) {
      return(cleaned_numeric)
    } else {
      return(NULL)
    }
  } else {
    numeric_result <- as.numeric(string_numeric)

    if (!is.na(numeric_result)) {
      return(numeric_result)
    } else {
      return(NULL)
    }
  }
}
```

```
date_achizitii$`Valoare (lei, fără TVA)` <- sapply(date_achizitii$`Valoare (lei, fără TVA)`,
curatare_numere)
```

Această funcție are rolul de a standardiza/curăța valorile numerice de potențiale greșeli de format, cum ar fi spre exemplu "100.85,20", acesta nu ar putea fi introdus vreodată în baza de date sub forma aceasta, însă, dacă aceste e trecut prin funcție acel "." va fi eliminat, iar acea "," va fi transformată într-un punct, căpătând astfel un format acceptabil.

```
calculate_data_finala <- function(data_initiala, data_finala) {
  #data_initiala <- clean_and_convert_date(data_initiala)
  data_finala <- tolower(data_finala)
  is_luni <- grepl("luni", data_finala)
  is_zile <- grepl("zile", data_finala)
  if (!is_luni && !is_zile) {
    return(data_finala)
  }
  num <- as.numeric(gsub("\\D", "", data_finala))
```

```

data_iniciala <- as.Date(data_iniciala, format = "%d.%m.%Y", tryFormats =
c("%Y-%m-%d", "%d.%m.%Y"))
result_date <- ifelse(is_luni, as.character(data_iniciala + months(num)),
                      ifelse(is_zile, as.character(data_iniciala + days(num)),
as.character(data_finala)))
return(result_date)
}

```

Această funcție are menirea de a converti date finale cu un format de tipul "x zile" sau "x luni" în date reale. Pentru început aceasta caută să vadă dacă argumentul conține "luni" sau "zile", în cazul în care nu conține, atunci funcția se oprește și returnează argumentul așa cum era el inițial, în caz contrar se va face suma dintre data inițială și numărul de zile sau luni descris în parametru și va fi returnat.

```

> # Test
> calculate_data_finala('17.01.2012', '13 zile' )
[1] "2012-01-30"

```

Figura 1.27: Exemplu de conversie a datei

Trebuie adusă în discuție tabela furnizori, aceasta a fost generată în R, importând coloana furnizori în serverul de SQL. Am vrut să creez un query mai complex, în cadrul serverului de SQL, care să elimine valorile cu sintaxa foarte apropiate (ex: 'SC FIRMA FIRMESCU' și 'S.C FIRMA FIRMESCUU'), de unde rezultă următoarea secvență de cod SQL.

```

select count(id_furnizor) from furnizori;
DELETE f1
FROM furnizori f1
JOIN furnizori f2
ON CHARINDEX(SUBSTRING(f2.denumire_furnizor, 1,
CASE
  WHEN LEN(f2.denumire_furnizor) > 99 THEN 15
  ELSE 20
END
), f1.denumire_furnizor) > 0
WHERE f1.denumire_furnizor <> f2.denumire_furnizor;
SELECT * INTO furnizori_backup FROM furnizori;
DROP TABLE furnizori;
CREATE TABLE furnizori
(
  id_furnizor int identity(1,1) primary key,
  denumire_furnizor varchar(250)
);
INSERT INTO furnizori (denumire_furnizor)

```

```
SELECT denumire_furnizor FROM furnizori_backup;
DBCC CHECKIDENT ('furnizori', RESEED, 0);
DROP TABLE furnizori_backup
select count(id_furnizor) from furnizori;
```

Pentru a putea converti denumirea funcțiilor cu id-urilor lor în R, este necesară importarea unei funcții SQL, care pe baza argumentului pe care îl primește, returnează id-ul celui mai apropiat 'match'.

```
dbhandle <- odbcDriverConnect('driver={SQL
Server};server=;database=Proiect_SSD;trusted_connection=true')

if (is.null(dbhandle)) {
  stop("Database connection failed.")
}

furnizorPatterns <- date_achizitii$`Denumire furnizor / prestator / executant`
furnizori <- numeric(length(furnizorPatterns))
for (i in seq_along(furnizorPatterns)) {
  furnizorPattern <- paste0("", furnizorPatterns[i], "")
  query <- paste("select dbo.RETURN_FURNIZOR_ID(", furnizorPattern, ")", sep = "")
  result <- sqlQuery(dbhandle, query)
  furnizori[i] <- result
  num_nulls <- sum(is.na(furnizori))
  num_nulls
}

odbcClose(dbhandle)
```

În cadrul codului de mai sus este apelată funcția din serverul sql pentru fiecare element din denumire furnizor, iar rezultatul este returnat în interiorul vectorului furnizori.

```
dbhandle <- odbcDriverConnect('driver={SQL
Server};server=;database=Proiect_SSD;trusted_connection=true')
for (i in 1:nrow(date_achizitii)) {
  id_sucursala_query <- paste("SELECT id_sucursala FROM sucursale WHERE
denumire_sucursala = ", date_achizitii$Sucursala[i], "", sep="")
  id_sucursala_fetch <- as.numeric(sqlQuery(dbhandle, id_sucursala_query))

  query <- paste("INSERT INTO Achizitii (id_sucursala, id_furnizor, data_achizitie,
data_finalizare, obiect, valoare, stadiu_executie, tip_procedura) VALUES (",
  id_sucursala_fetch, ", ",
```



```

furnizori[i], ", ",
"", date_achizitii$Data_Începere[i], ", ",
"", date_achizitii$`Dată finalizare/Timp necesar`[i], ", ",
"", date_achizitii$Obiect[i], ", ",
"", date_achizitii$`Valoare (lei, fără TVA)`[i], ", ",
"", date_achizitii$`Stadiu execuție (%)`[i], ", ",
"", date_achizitii$`Tipul procedurii aplicate pt. atribuire`[i], """),
sep="")

tryCatch({
  odbcQuery(dbhandle, query)
}, error = function(e) {
  cat("Error: ", e$message, "\n")
})
}

odbcClose(dbhandle)

```

În această ultima secvență de cod este apelată inițial o interogare care returnează id-ul unei sucursale pe baza denumirii acesteia. Apoi are loc exportarea elementului dataframe ului în SQL.

1.3 Analiză descriptivă scurtă a datelor în R

Am realizat o scurtă analiză descriptivă în vedere mai bune înțelegeri a datelor. Variabilă vizată o reprezintă valoarea achiziției.

```

> summary(date_achizitii$`valoare (lei, fără TVA)`)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
   132   10647   30836  116095   88162 53970904      81
> summary(date_achizitii$`valoare (lei, fără TVA)`[date_achizitii$Sucursala == 'Bucuresti'])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
   172   24406   67990  259750  156308 53970904      6
> summary(date_achizitii$`valoare (lei, fără TVA)`[date_achizitii$Sucursala == 'Craiova'])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1080  10208  29782 121497  71264 7299416
> summary(date_achizitii$`valoare (lei, fără TVA)`[date_achizitii$Sucursala == 'Timisoara'])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   139  13270  33300 101848  89020 1441747
> summary(date_achizitii$`valoare (lei, fără TVA)`[date_achizitii$Sucursala == 'Brasov'])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
   623  12270  31256  99637 101367 1084740      1
> summary(date_achizitii$`valoare (lei, fără TVA)`[date_achizitii$Sucursala == 'Iasi'])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
 131.8  7951.2 23205.5  87035.6  73787.5 3003238.0      3
> summary(date_achizitii$`valoare (lei, fără TVA)`[date_achizitii$Sucursala == 'Galati'])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   737   7526  19338  56620  47474 3153600
> summary(date_achizitii$`valoare (lei, fără TVA)`[date_achizitii$Sucursala == 'Constanta'])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
   300   9939  23599  56149  53188 1000371      1

```

Figura 1.28: Statistici descriptive valoare

Se poate observa cum valoare are o variabilitate extrem de mare pentru fiecare sucursală în parte.

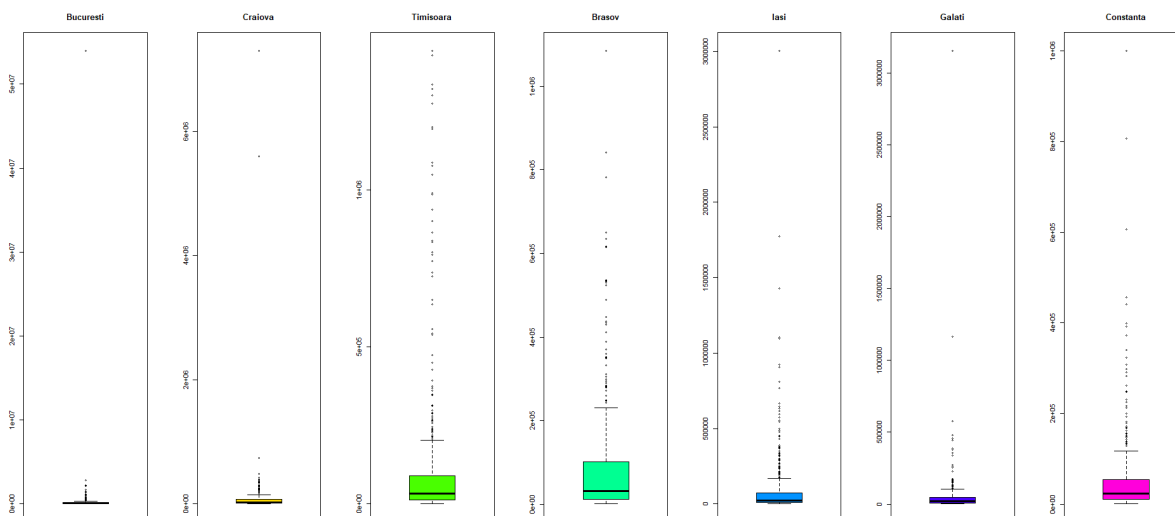


Figura 1.29: Boxplot valoare

În figura de mai sus se poate observa că sucursalele înregistrează outlieri pentru valorile superioare medianei, aceștia pot fi explicați de natură neechilibrată a achizițiilor necesare, costul nu contează atât de mult ca utilitatea adusă.

Incidența valorilor outlier este prea mare pentru a putea fi afișată integral în documentația proiectului, însă am atașat o parte mică din ea în figura de mai jos.

```
Sucursala: Bucuresti
Outliers însoțiți de data în care s au produs:
outlier: 441000 | Date: 04 Jan 2016
outlier: 492000 | Date: 21 Jan 2016
outlier: 53970904 | Date: 27 Jan 2016
outlier: 596160 | Date: 28 Jan 2016
outlier: 372713 | Date: 01 Feb 2016
outlier: 537043 | Date: 17 Feb 2016
outlier: 494992.8 | Date: 08 Mar 2016
outlier: 430267.1 | Date: 10 Mar 2016
outlier: 433347.6 | Date: 17 Mar 2016
outlier: 374542.1 | Date: 14 Apr 2016
```

Figura 1.30: Exemplu incidență outlieri

În figura de mai jos se poate observa frecvența valorilor fiecărei sucursale, fiecare dintre acestea prezintă o distribuție puternic leptocurtica cu asimetrie foarte pronunțată la dreapta, așadar există diferențe extreme între frecvențe, acestea având sa se cumuleze într-un spațiu foarte restrâns în partea stângă a distribuției

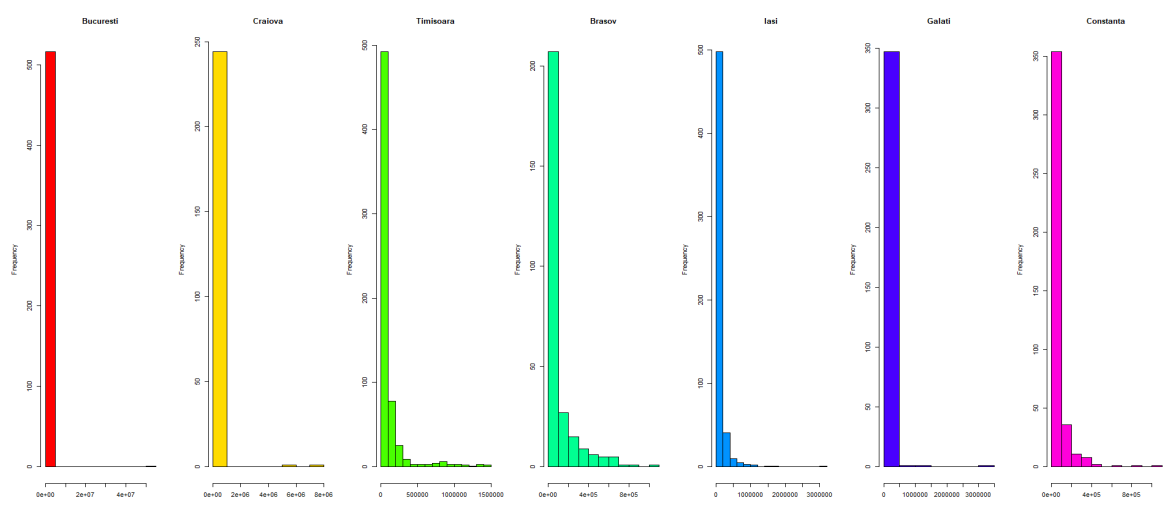


Figura 1.31: Frecvența valorilor sucursalelor

Partea 2 - Modelarea și analiza datelor

2.1 Analiza cu Charts și Dashboard în Power BI

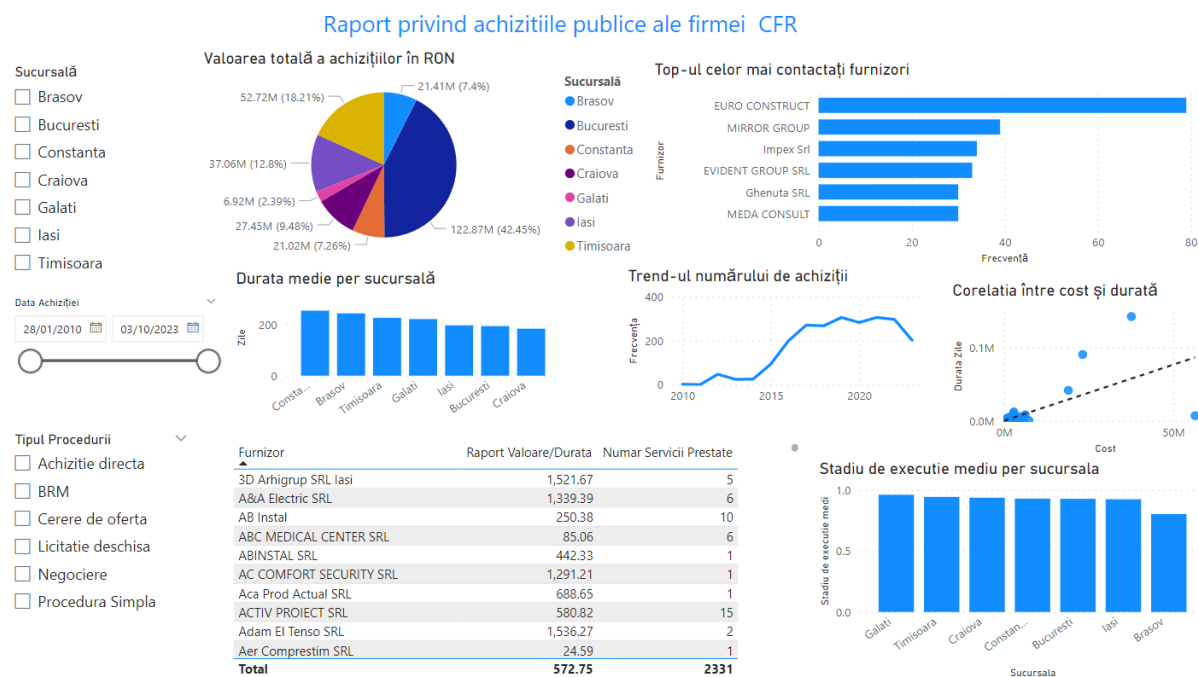


Figura 2.1: Dashboard Achiziții

În acest dashboard sunt surprinse informații cantitative cu privire la datele din tabela Achiziții, în materie de valoare, durată, eficiență, etc. De asemenea sunt prezente slidere pentru filtrarea în funcție de Sucursală, Tipul procedurii și data achiziției.

Din dashboard reies următoarele concluzii:

- București generează cele mai mari cheltuieli, sucursala fiind responsabilă pentru aproape jumătate dintre toate achizițiile firmei.
- Cea mai puțin eficientă sucursală este Brașov, achizițiile efectuate de sucursală având în medie un stadiu de execuție de doar 80%
- Numărul achizițiilor continuă să crească de la an la an
- Durata medie ce a mai mare a unei achiziții este în sucursala Constanța, iar cea mai mică este în Brașov
- Între Cost și Durată există o corelație destul de semnificativă când ambele au valori scăzute, însă atunci când se înregistrează outlieri în una dintre cele două, valorile tind preponderent să nu coreleze între ele.

Raport privind angajatii firmei CFR

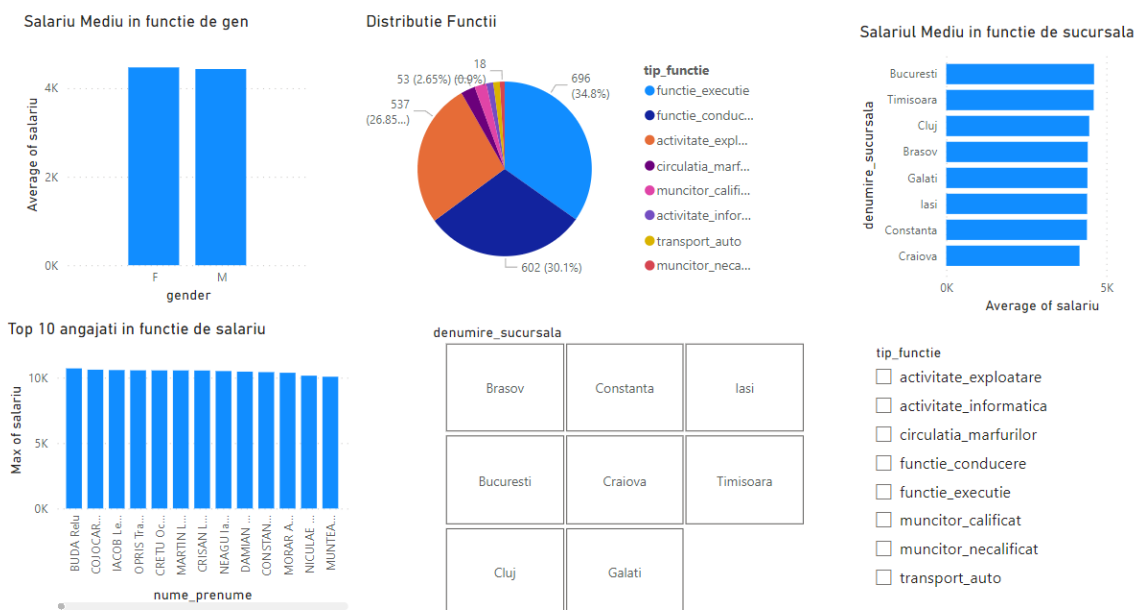


Figura 2.2 : Dashboard Angajați

De asemenea, am realizat un mic dashboard și pentru angajați, însă după cum se poate observa datorită naturii simulării, variabilitatea datelor este redusă.

Se pot observa următoarele lucruri:

- Nu există aproape nicio diferență între plata primită de femei și bărbați
- Funcțiile de tip executiv, conducere și activitate de exploatare ocupă 90% din ponderea funcțiilor angajaților

- Salariul Mediu este relativ similar în toate sucursalele, însă cea mai mare valoare este 4614 (înregistrată în București), iar cea mai mică 4161(înregistrată în Craiova)
- Primii 10 angajați au o valoare similară foarte apropiată. Diferența salarială între primul angajat și cel de-al zecelea este mai mică de 10%
- Per total, în cazul acestei simulări, locul de muncă pare unul foarte sănătos

2.2 Prognoza unor indicatori micro- sau macro economici

Pentru această etapă, am decis sa efectuez prognoza cifrei de afaceri, deoarece este un indicator semnificativ în evaluare performanței unei firme, evoluția sa determinând potențialul firmei pe piață.

Am extras acest indicator din bilanțul contabil al firmei, care este public și vizează datele din primul semestru al anilor 2012-2023. Am extins orizontul de previziune până în anul 2024.

Mediul în care am decis să realizez această prognoză este Excel, iar metodele folosite sunt ”three months average” și aplicarea regresiei (folosind funcția trend() din Excel)

Indice Perioadă	Perioada	Cifra de afaceri netă
0	SEM 1 -2012	763075.59
1	SEM 1 -2013	779022.45
2	SEM 1 -2014	970152.21
3	SEM 1 -2015	897800.54
4	SEM 1 -2016	816926.65
5	SEM 1 -2017	931864.82
6	SEM 1 -2018	868640.36
7	SEM 1 -2019	883464.23
8	SEM 1 -2020	760971.09
9	SEM 1 -2021	811651.87
10	SEM 1 -2022	939373.33
11	SEM 1 -2023	1325048.35

Figura 2.3: Date inițiale cifră de afaceri

Pe baza acestor date, urmează să fie aplicate 3 months average și metoda regresiei, iar apoi să fie măsurate tipurile de erori, pentru a vedea care dintre aceste metode, ar fi cea mai stabilă pentru viitoarele previziuni.

3 month moving Average folosit pe datele istorice	TREND() folosit pe date istorice	Eroare 3 month moving average				Eroare TREND()			
		Error	ABS	Error %	Sq.Error	Error	ABS	Error %	Sq.Error
763075.59	763075.59	0.00	0	0%	0.00	0.00	0	0%	0.00
779022.45	779022.45	0.00	0	0%	0.00	0.00	0	0%	0.00
970152.21	970152.21	0.00	0	0%	0.00	0.00	0	0%	0.00
837416.75	1044493.37	-60383.79	60383.79	-7%	3646202416.81	146692.82	146692.8	14%	21518784710.89
862197.14	1148031.67	-45270.48	45270.48	6%	2049416610.93	331105.02	331105	29%	109630535483.25
889922.03	1251569.98	-41942.78	41942.78	-5%	1759197154.53	319705.17	319705.2	26%	102211394659.05
863178.64	1355108.29	-5461.72	5461.72	-1%	29830389.67	486467.93	486467.9	36%	236651049675.14
871765.94	1458646.60	-11698.29	11698.29	-1%	136850105.23	575182.37	575182.4	39%	330834754540.81
874955.53	1562184.91	113984.44	113984.4	15%	12992453651.61	801213.82	801213.8	51%	641943588296.78
869966.70	1665723.22	58314.83	58314.83	7%	3400619724.99	854071.35	854071.3	51%	729437866335.78
872229.39	1769261.53	-67143.94	67143.94	-7%	4508308964.66	829888.19	829888.2	47%	688714412604.18
872383.88	1872799.83	-452664.47	452664.5	-34%	204905123168.64	547751.49	547751.5	29%	300031691875.88

Figura 2.4 : Tabel prognoze și erori generate

În figura de mai sus sunt afișate datele în urma aplicării celor 2 metode și calculării erorilor.

ME	-46858	ME	543564
MAE	95207	MAE	543564
MAPE	-3%	MAPE	36%
MSE	25936444687	MSE	351219342020

Figura 2.5: Erorile Totale

În figura de mai sus sunt afișate cele 4 tipuri ale erorii, în dreapta pentru 3 months average, iar în stânga pentru metoda regresiei, realizată prin intermediul funcției trend().

Prin interpretarea coeficientului MAPE (eroarea medie procentuală), putem spune că în medie, metoda three months a prezis valori cu -3% mai mici decât în realitate, iar metoda regresiei a prezis valori cu 36% mai mari ca în realitate, așadar putem spune cu ușurință că metoda "three months average" a fost mult mai practică, lucru care se poate vedea și comparând coeficienții MSE (eroarea medie pătratică - cu cât aceasta are o valoare mai mare cu atât mai prevalente/semnificative sunt erorile), cea de-a doua metodă având un coeficient de peste 10 ori mai mare ca prima.

	Indice Perioadă	Perioada	Cifra de afaceri netă	3 month Moving Average	TREND()
Orizont de prognoza	12	SEM 1 -2024	?	1025357.85	1010427.44
	13	SEM 1 -2025	?	1096593.18	1031293.16
	14	SEM 1 -2026	?	1148999.79	1052158.88
	15	SEM 1 -2027	?	1090316.94	1073024.61
	16	SEM 1 -2028	?	1111969.97	1093890.33
	17	SEM 1 -2029	?	1117095.57	1114756.05
	18	SEM 1 -2030	?	1106460.82	1135621.78
	19	SEM 1 -2031	?	1111842.12	1156487.50
	20	SEM 1 -2032	?	1111799.50	1177353.22
	21	SEM 1 -2033	?	1110034.15	1198218.95
	22	SEM 1 -2034	?	1111225.26	1219084.67

Figura 2.6: Date numerice previzionate

Pe baza acestor date putem realiza reprezentări grafice ale trendului pentru a evidenția acuratețea fiecărei previziuni.

În acest scop, am realizat două grafice, unul care urmărește trend-ul ”previziunilor” folosite pe datele istorice și încă unul care urmărește potențiala evoluție a orizontului de prognoză.

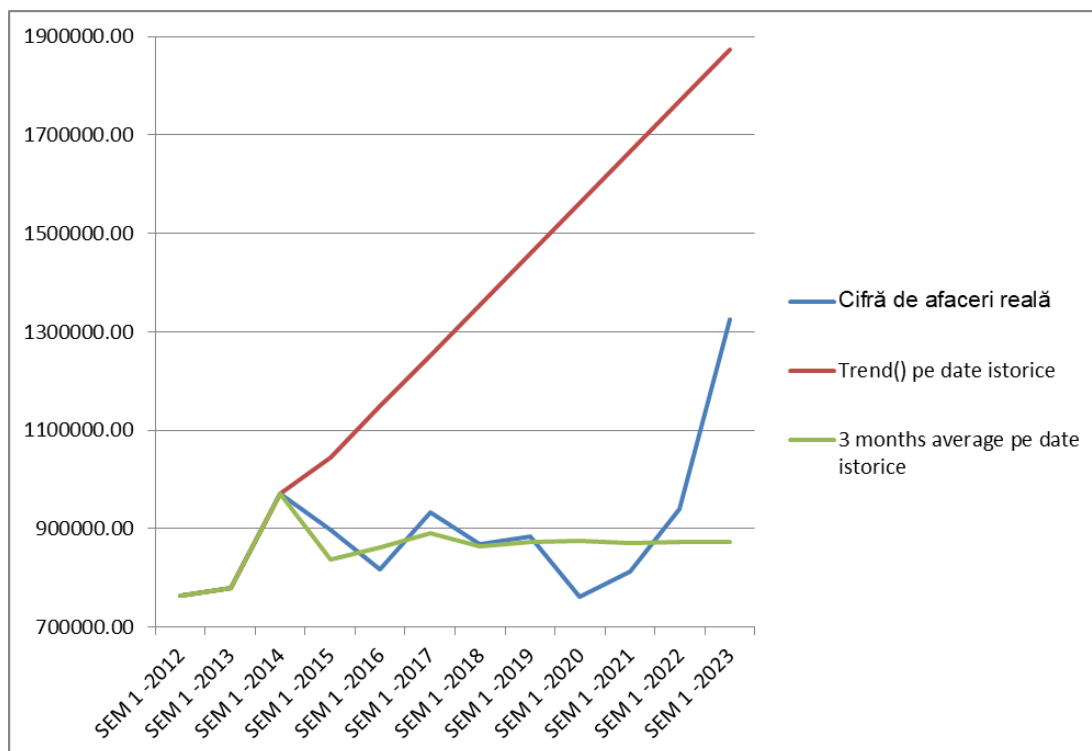


Figura 2.7: Comparație între realitate și cele două prognoze

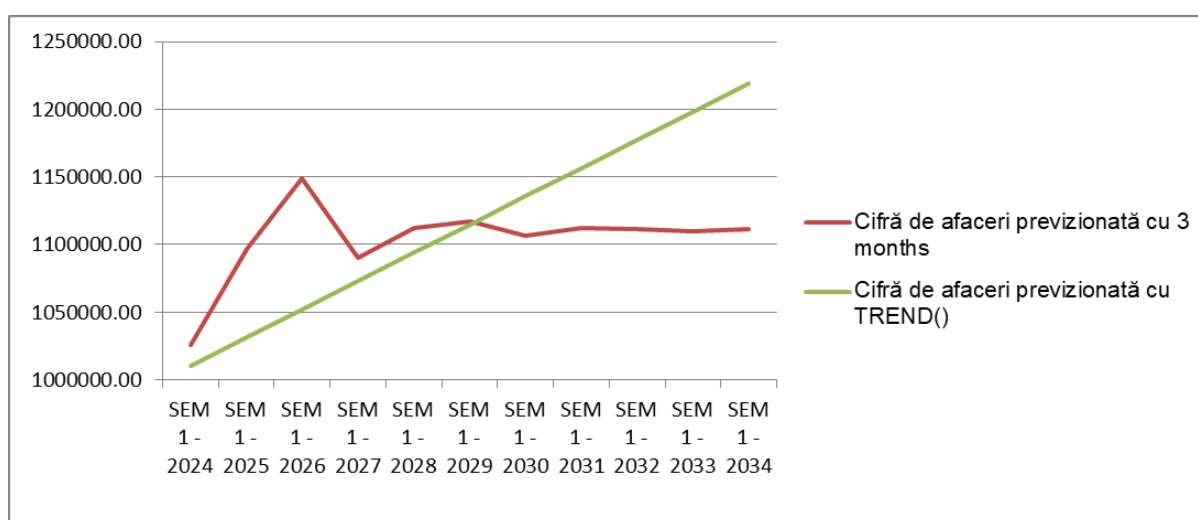


Figura 2.8: Potențialul trend al orizontului de prognoză

Analizând figura 2.8, ni se prezintă două scenarii ipotetice:

1. În cazul previziunii cu 3 months average, se observă o creștere pe orizontul primilor 3 ani, urmată de o scădere în următorul an, urmată de o perioadă de stagnare consistentă, până la sfârșitul perioadei prognozate, valoare finală fiind totuși mai mare decât cea inițială cu aproximativ 10% (1000).
2. În cazul regresiei, trendul este unul strict liniar crescător valoare cifrei de afaceri crescând cu aproximativ 2,5% în fiecare an, ajungând la o valoare mai mare cu peste 20% (2000) decât cea inițială.

Ambele scenarii au impact pozitiv asupra firmei, dar 3 months average, reda un viitor mai pesimist, indicând stagnarea firmei pe termen lung, în timp ce regresia arată un viitor optimist, indicând o creștere constantă.

2.3 Rezolvarea unei probleme decizionale economice în cadrul respectivei companii

În cadrul acestui capitol, mi-am propus să realizez o scurtă analiză text cu Ant Conc, dar și să implementez lanțurile lui Markov în Excel.

Enunț:

Să se identifice care au fost motivele achizițiilor publice ale firmei CFR, apoi să se vizualizeze datele obținute. După aceea se reclasifice ponderile obținute, după 3 macro-categorii: Menținerea, Servicii, Achiziții și să calculeze și reprezinte grafic evoluția ponderilor pe o durată de 5 unități de timp (ani), având la dispoziție matricea probabilităților de trecere

Obiectiv	Reorientari de fonduri catre		
	Ch Ment	Ch Servicii	Ch Achiz
Cheltuieli Menținenta	0.200	0.400	0.400
Cheltuieli Servicii	0.400	0.250	0.350
Cheltuieli Achizitii	0.400	0.350	0.250

Figura 2.9: Matricea Probabilităților de trecere

Etapa 1:

Pentru început trebuie extrasă coloana obiectiv din data frame-ul construit anterior sub format text pentru a putea astfel importa datele în ANTCONC.

```
writeLines(date_achiziții$Obiect, "objective.txt")
```


Etapa 2:

Se crează un nou corpus și se adaugă fișierul text, creat în R, apoi se selectează ca target corpus. Apoi se selectează coloana Word => Adv Search => Search Query List și se introduc cuvinte relevante analizei. Pentru a deduce ce cuvinte ar putea fi prevalente există posibilitatea de a viziona și a dat scroll întregului document prin intermediul ferestrei "File View"

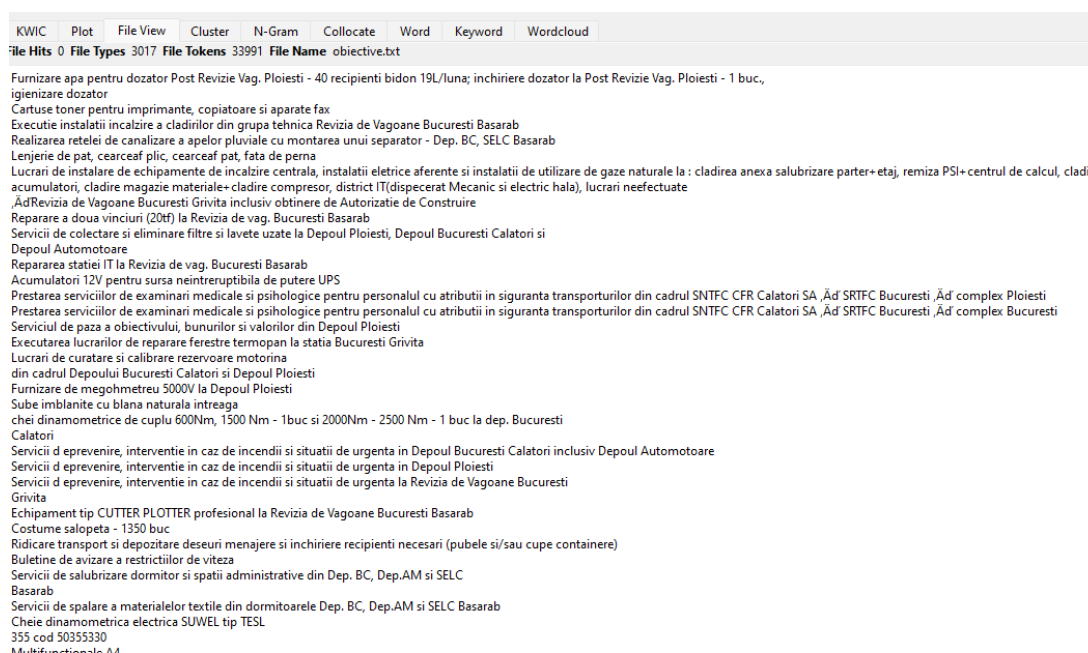


Figura 2.10: Fereastra File View

De asemenea se pot accesa N-Gram sau Word și apoi să se realizeze o căutare fără termeni specifici pentru a afișa cele mai folosite cuvinte sau grupe de cuvinte.

După stabilirea cuvintelor pe care le vrem găsite, se accesează Word și se adaugă în Query List, apoi se realizează o căutare.

	Type	Rank	Freq	Range
1	servicii	1	793	1
2	salubritare	2	313	1
3	reparatii	3	238	1
4	intretinere	4	224	1
5	apa	5	132	1
6	personal	6	109	1
7	transport	7	91	1
8	paza	8	89	1

Figura 2.11: Output parțial din ANTCOCONC

Etapa 3:

Se importă datele în EXCEL, păstrându-se coloanele TYPE, RANK(opțional) și FREQ, se crează și o coloană nouă, care să indice categoria obiectului, această coloană poate conține doar categoriile stabilite de decident.

Type ▼	Rank ▼	Freq ▼	Categorie ▼
salubritate	2	313	mentenanta
reparatii	3	238	mentenanta
intretinere	4	224	mentenanta
apa	5	132	combustibil
personal	6	109	personal
paza	8	89	securitate
lenjerie	9	86	bunuri de consum

Figura 2.12: Tabel analiză text Excel

Etapa 4:

Se realizează un tabel în care este însumată frecvența fiecărei categorii, iar apoi se calculează procentul ocupat de fiecare din ponderea totală, pe baza acestora se realizează un pie chart.

Total Relevant Freq	1962	Percentages
mentenanta	931	47.5%
personal	178	9.1%
securitate	195	9.9%
bunuri de consum	195	9.9%
analiza	87	4.4%
combustibil	226	11.5%
modernizare/inovare	44	2.2%
componente feroviale	106	5.4%

Figura 2.13: Frecvențe și procentaje

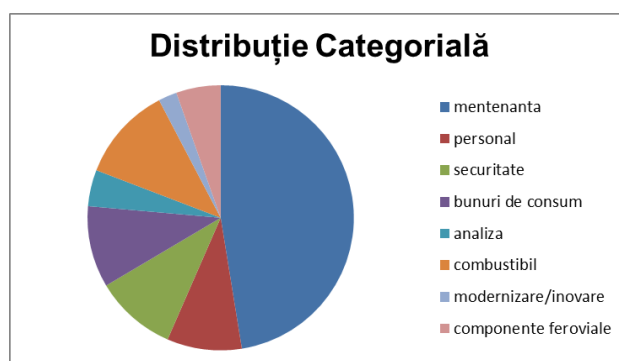


Figura 2.14: Distribuție Categoricală

Etapa 5:

Realizăm redistribuirea ponderilor în funcție de cele 3 macro categorii stabilite în enunț.

Cateogori Generalizate	Procentaj
Cheltuieli Mentenanta	0.475
Cheltuieli Servicii	0.234
Cheltuieli Achizitii	0.291

Figura 2.15: Ponderi macro categorii

Ne folosim de funcția MMULT() pentru a calcula produsul dintre matricea probabilităților de trecere și vectorul de macro categorii de mai sus.

	Ch Ment	Ch Serv	Ch Achiz
T0	0.474516	0.234455	0.29103
T1	0.305097	0.35028	0.344623
T2	0.338981	0.330227	0.330793
T3	0.332204	0.333926	0.33387
T4	0.333559	0.333218	0.333223
T5	0.333288	0.333356	0.333356

Figura 2.16: Prognoză macro categorii

În concluzie, pe baza prognozei realizate, procentul Mentenanței va scădea de la 47% la 33% în 5 ani, iar cel Serviciilor va crește de la 23% la 33% și cel al Achizițiilor de la 29% la 33%, de altfel putem observa că va avea loc o situație de echilibru perfect, în care toate cele 3 macro categorii au aproximativ aceeași pondere.

Etapa 6:

Realizăm un line chart pe baza datelor obținute.

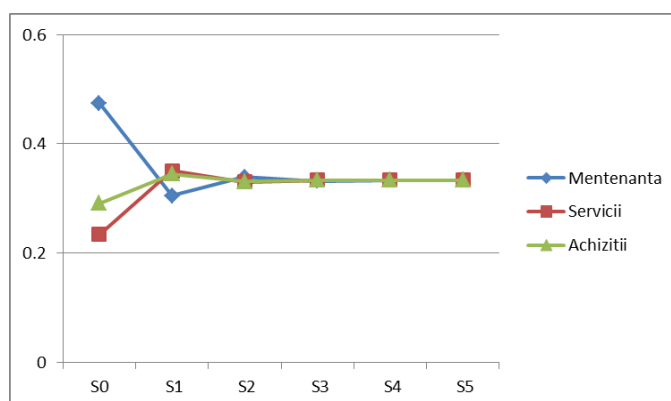


Figura 2.17: Line chart ponder

Pe baza line chart-ului efectuat, putem observa că toate ponderile au tendința de a se apropia de echilibru, cea mai mare schimbare se petrece la trecerea din S0 în S1, iar apoi, treptat, macro categoriile se apropie de 33%.

Partea 3 - Interfața cu utilizatorul

În vederea prezentării interfețelor cu utilizatorul al sistemelor suport de deciziei folosite, am atașat următoarele print screen-uri:

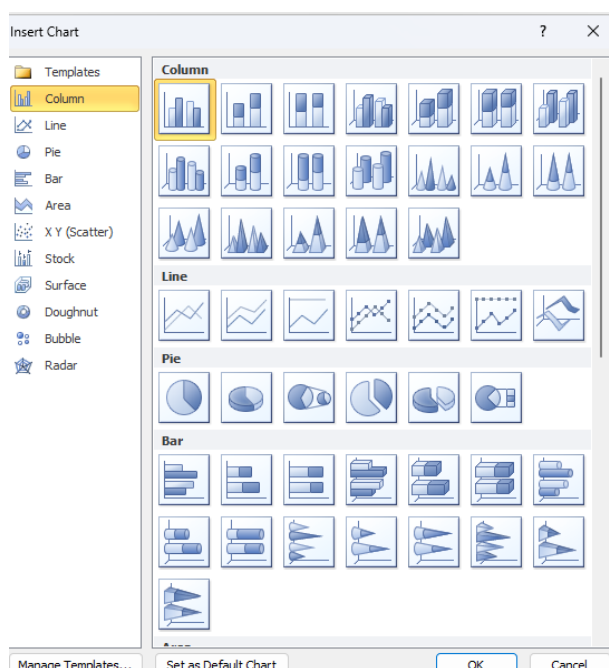


Figura 3.1: Inserare Chart Excel

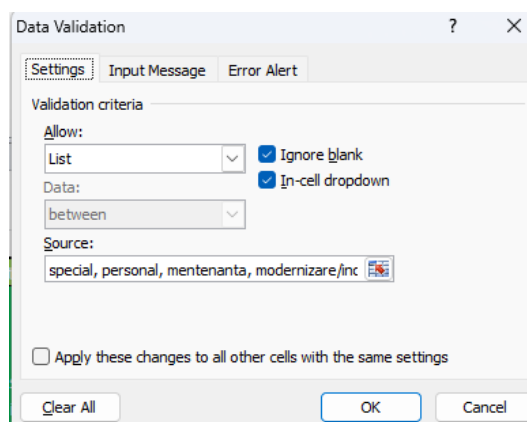


Figura 3.2: Validare Date Excel

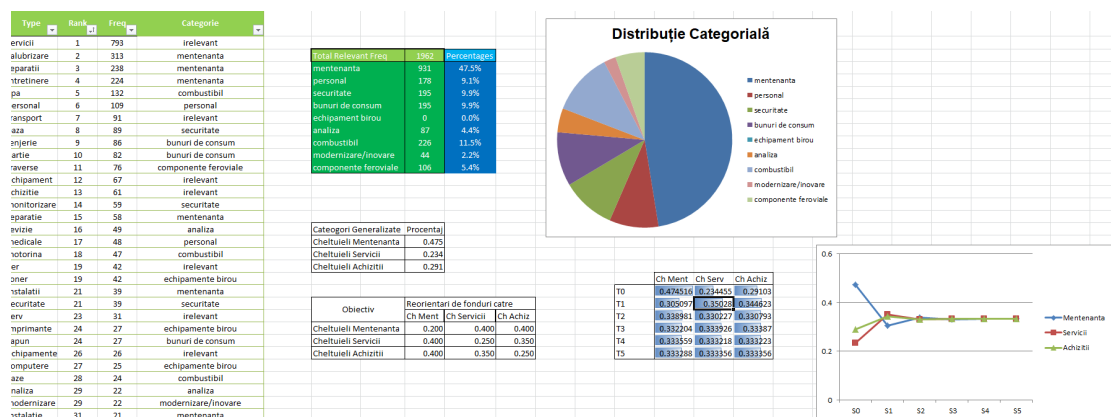


Figura 3.3: Aspectul unei foi de calcul Excel

```

Script_proiect_SSD.R
56
57 numere_telefon_generate <- character(2000)
58
59 for (i in 1:2000) {
60   random_digits <- sample(0:9, size = 8, replace = TRUE)
61   numere_telefon <- paste0("07", paste0(random_digits, collapse = ""))
62   numere_telefon_generate[i] <- numere_telefon
63 }
64
65 date_angajati <- data.frame(
66   id = id,
67   nume_generate = nume_generate,
68   prenume_generate = prenume_generate,
69   genuri = genuri,
70   emails_generate = emails_generate,
71   numere_telefon_generate = numere_telefon_generate,
72   sucursale_generate = sucursale_generate,
73   functii_generate = functii_generate
74 )
75
76 date_angajati$salariu_mediu_values <- unlist(salariu_mediu_values)
77
78 write.csv(date_angajati, "date_angajati.csv", row.names = FALSE)
79
80 dbhandle <- odbcDriverConnect('driver={SQL Server};server=DESKTOP-TTIER45;database=Proiect_SSD;trusted_connection=true')
81
326:1 Prelucrare date achizitii pentru export-ul catre baza de date
R Script

```

Figura 3.4: Aspectul unui Fișier deschis în Rstudio

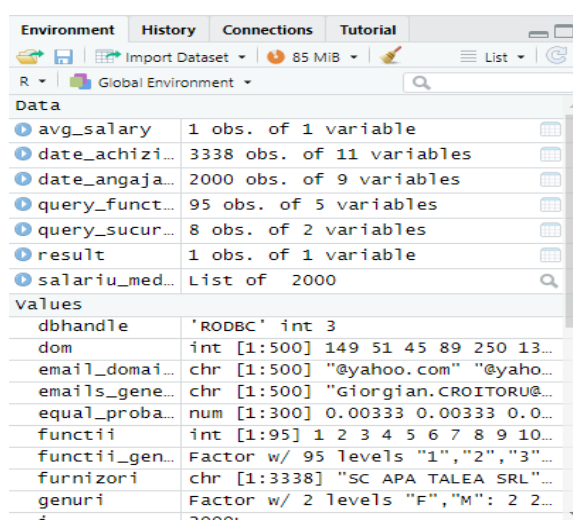


Figura 3.5: Aspectul secțiunii Environment în Rstudio

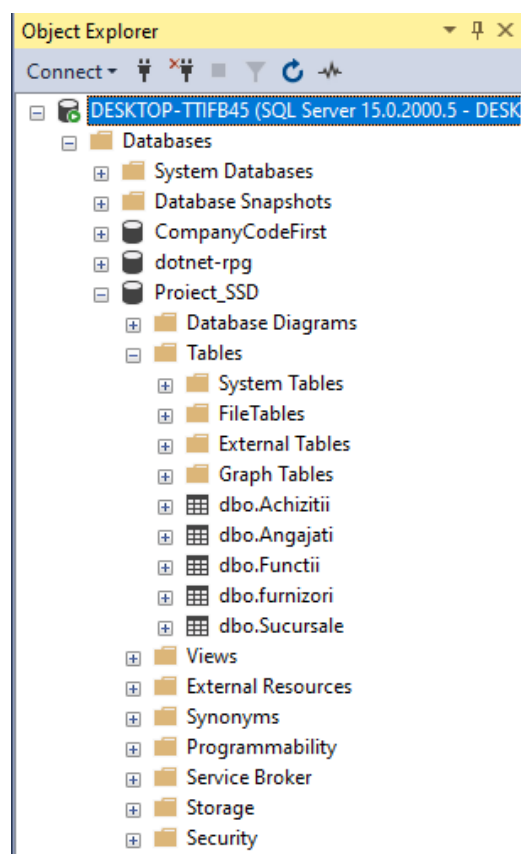


Figura 3.6: Object Explorer în SQL SERVER

DESKTOP-TTIFB45.Pr...SSD - dbo.Achizitii			
Column Name	Data Type	Allow Nulls	
id_achizitie	int	<input type="checkbox"/>	
id_sucursala	int	<input checked="" type="checkbox"/>	
data_achizitie	date	<input checked="" type="checkbox"/>	
data_finalizare	date	<input checked="" type="checkbox"/>	
obiect	varchar(MAX)	<input checked="" type="checkbox"/>	
valoare	float	<input checked="" type="checkbox"/>	
durata_zile	int	<input checked="" type="checkbox"/>	
tip_procedura	varchar(MAX)	<input checked="" type="checkbox"/>	
id_furnizor	int	<input checked="" type="checkbox"/>	
stadiu_executie	decimal(5, 2)	<input checked="" type="checkbox"/>	

Figura 3.7: Vizualizarea structurii unui tabel în SQL SERVER

AntConc

File Edit Settings Help

Target Corpus
Name: objective_achizi
Files: 1
Tokens: 33991
objective.txt

KWIC Plot File View Cluster N-Gram Collocate Word Keyword Wordcloud

Collocate Types 18 Collocate Tokens 144 Page Size 100 hits 1 to 18 of 18 hits

	Collocate	Rank	FreqLR	FreqL	FreqR	Range	Likelihood	Effect
1	legare	1	9	0	9	1	58.391	6.030
2	pamant	2	9	0	9	1	54.300	5.709
3	scb	3	6	0	6	1	42.448	6.446
4	prize	4	6	6	0	1	37.439	5.861
5	lfi	5	6	0	6	1	27.306	4.638
6	interioare	6	5	2	3	1	21.944	4.520
7	verificare	7	11	10	1	1	20.434	2.513
8	utilizare	8	5	0	5	1	18.930	4.067
9	termice	9	5	0	5	1	17.896	3.909
10	reparatii	10	12	11	1	1	17.250	2.136
11	elctrice	11	2	1	1	1	17.102	7.446
12	administrativa	12	4	2	2	1	16.680	4.358
13	intretinere	13	11	11	0	1	15.344	2.098
14	distributie	14	3	3	0	1	15.234	5.030
15	naturale	15	4	3	1	1	15.062	4.053
16	constructii	16	3	3	0	1	14.537	4.861
17	gaze	17	4	3	1	1	14.052	3.861
18	si	18	39	29	10	1	13.016	0.886

Search Query ☒ Words ☐ Case ☐ Regex Window Span From 5L To 5R Min. Freq 1 Min. Range 1

Instalații

Sort by Likelihood ☐ Invert Order

Figura 3.8: Collocate pentru cuvântul instalații - ANTCONC

Corpus Source

☒ Corpus Database ☐ Raw File(s) ☐ Word List

Corpus Database Library

Add Database File(s) Add Database Dir Export Library Import Library

View Expand level 1 Show online corpora

Available Corpora (db)	Status	Corpus Size
Default (List)		
AmE06 (List)		
BE06 (List)		
demo.db	Ready	1 MB
User (List)		
date_achiziti.db	Ready	6 MB
objective_achiziti.db	Ready	9 MB
temp.db	Ready	36 KB

Target Corpus Reference Corpus

Corpus name objective_achiziti.db Swap (with reference)

Description

Category	Description
full_name	objective_achiziti
short_name	objective_achizi
file_count	1
token_count	33991
type_count	3017
encoding	mac_latin2
token_definition	[^\s!]+
ignore_header	False
ignore_items	False
number_replace	False
format	raw_files
indexer_type	type
indexer	simple_word_indexer

Figura 3.9: Corpus Manager - ANTCONC

AntConc - Advanced Search

☒ Search Query List

Search Query ☒ Words ☐ Case ☐ Regex

testare

achizitie

achizitii

aer

ajutor

Drag and drop or paste search queries into the box above

Figura 3.10: Advanced Search - ANTCONC

Partea 4- Managementul cunoștințelor

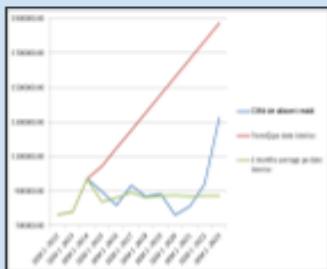
Pe baza analizelor efectuate, am obținut segmentarea cheltuielilor publice a firmei, scenarii de evoluție a cifrei de afaceri și a cheltuielilor cu achizițiile publice, cunoașterea eficienței sucursalelor în vederea folosirii fondurilor pentru achiziții publice, identificarea celor mai de încredere furnizori, o idee de ansamblu cu privire la ponderea alocării fondurilor în fiecare sucursală, dar și cum ar trebui să arate repartitia funcțiilor angajaților și salariile acestora într-o simulare lipsită de bias.

Schema Sistemului Suport de Decizie:



Managementul Bazei de Date

- SQL SERVER: Creare tabele, legături, funcții și interogări
- RSTUDIO: Standardizare, curățare și populare tabele
- Excel: Standardizare și Curățare a datelor



Modelare Și Analiză

- EXCEL: Realizarea de prognoze și vizualizarea datelor
- RSTUDIO: Realizarea unei analize descriptive a datelor și vizualizarea datelor
- POWER BI: Realizarea de dashboard-uri
- ANTCONC: Analiză Text



Interfața cu utilizatorul

- SQL SERVER: Lucrul cu baza de date
- RSTUDIO: Mediu de programare
- EXCEL: Foi de calcul
- POWER BI: Dashboard
- ANTCONC: Analiză de text



MANAGEMENTUL CUNOȘTINȚELOR

- Scenarii de Evoluție
- Scenarii de Optimalitate
- Monitorizarea Eficienței
- Cunoașterea distribuției cheltuielilor
- Cunoașterea punctelor tari și slabe

Concluziile analizei

În urma realizării acestei analize, am dezvoltat numeroase cunoștințe pe baza datelor făcute publice de către organizația ”Căile Ferate Române”, mai precis, o bună înțelegere a modului în care această firmă își gestionează resursele și cât de bine o face.

Pentru realizarea acestui lucru a fost necesară o curățare și standardizare riguroasă a datelor, deoarece acestea aveau un caracter foarte haotic, acest proces de ”sanitizare” a datelor a făcut posibilă modelarea cu acuratețe a situației firmei, cât și a prognozei unor scenarii viitoare potențiale.

Prin urmare, am dovedit că implementarea Sistemului Suport de Decizie, facilitează și eficientizează procesul de modelare, vizualizare, calcul și luare a deciziilor.

Surse web pentru date

- <https://www.cfrcalatori.ro/achizitii-publice/>
- <https://www.cfrcalatori.ro/buget/>
- <https://www.cfrcalatori.ro/raportari-contabile/>