



Academia de Studii Economice, Bucureşti
Facultatea de Cibernetică, Statistică și Informatică Economică
Specializarea: Cibernetica Economică
Anul 2024-2025

Proiect Pachete Software

***Analiza activitatii si a eventualelor posibilitati de extindere ale
companiei Booking.com***

Proiect realizat de Iordan Maria-Alexandra, Grosu Catalina-Ionela

Cadru didactic coordonator: Asistent univ. Dr. Dobrita Gabriela

CUPRINS

Introducere	4
PARTEA 1 – PROGRAMARE PYTHON	5
1. Liste	5
2. Tupluri	6
3. Dictionare.....	8
4. Seturi	10
5. Definirea si apelarea functiilor, structuri conditionate si repetitive	11
6. Importul unui fisier CSV.....	13
7. Tratarea valorilor lipsa. Stergerea coloanelor si a inregistrarilor.....	15
8. Analiza datelor din fisierul CSV. Analiza descriptiva.....	18
9. Accesarea datelor cu LOC si ILOC	23
10. Gruparea si agregarea datelor.....	25
11. Reprezentari grafice	29
12. Prelucrarea seturilor de date utilizand MERGE	32
PARTEA 2 – PROGRAMARE SAS	33
1. Crearea unui set de date SAS din fisiere externe.....	33
2. Crearea si folosirea de etichete si formate definite de utilizator	35
3. Procesarea conditionala a datelor	37
4. Procesarea iterativa a datelor	39
5. Utilizarea functiilor SAS	42
6. Combinarea seturilor de date prin proceduri specifice SAS și SQL.....	44
7. Masive	45
8. Utilizarea de proceduri pentru raportare.....	47
8.1. Raport detaliat pe orase	47
8.2. Raport privind hotelurile din fiecare Stat	49
9. Proceduri statistice.....	50
9.1. Analiza univariată a prețurilor.....	50
9.2. Procedura MEANS	52
10. Generarea de grafice.....	54
11. Corelatii	56

11.1.	Corelatia dintre Rating si Pret.....	56
11.2.	Corelatia dintre numarul de recenzii si pret	57
11.3.	Procedura CORR.....	58
12.	Regresie.....	59
12.1.	Influenta rating-ului asupra pretului.....	59
12.2.	Influenta numarului de recenzii asupra pretului.....	61
13.	ANOVA.....	63
PARTEA 3 – SAS ENTERPRISE GUIDE		64
1.	Importul unui fisier Excel	64
2.	Interogari	67
2.1.	Selectarea coloanelor si filtrarea campurilor	67
2.2.	Crearea unei coloane calculate	68
2.3.	Crearea unei coloane recodificate	69
2.4.	Joinctiuni.....	71
2.5.	Interogari cu parametri.....	73
3.	Prelucrarea datelor	74
3.1.	Crearea unui raport lista.....	74
3.2.	Agregarea datelor	75
4.	Grafice.....	78
Realizarea de corelatii si grafice de tip scatter		80

Introducere

Industria hotelieră din Olanda este una extrem de dinamică, având un impact direct asupra turismului și economiei locale. **Booking.com**, fiind una dintre cele mai mari platforme de rezervare a unităților de cazare la nivel global, joacă un rol important în facilitarea accesului turiștilor la hoteluri din diverse regiuni ale țării.

În acest proiect, ne propunem să analizăm activitatea companiei Booking.com pe piața hotelieră olandeză, având ca scop înțelegerea tendințelor actuale, identificarea factorilor care influențează performanța hotelurilor și explorarea oportunităților de extindere. Această analiză este realizată utilizând limbajul de programare Python, aplicând tehnici de prelucrare și vizualizare a datelor pentru a obține insight-uri relevante.

Scopul acestui proiect este de a înțelege factorii care influențează performanța hotelurilor listate pe Booking.com și de a identifica posibile oportunități de extindere a companiei pe piața olandeză. Analiza se bazează pe un set de date ce conține informații despre hoteluri din diverse regiuni ale Olandei, utilizând pachetele de programe Python, SAS și SAS Enterprise Guide pentru prelucrarea și interpretarea datelor.

Setul de date, obținut de pe platforma Kaggle, include următoarele 9 coloane pentru analiza activității hotelurilor:

- **ID** – Identificator unic al fiecărui hotel din setul de date.
- **Name** – Numele fiecărui hotel listat pe platformă.
- **Place** – Locația (City & State) în care se află hotelul.
- **Type** – Tipul camerei disponibile (ex: Superior Double Room, 2-person Premium Hotelroom, City Double Room etc.).
- **Price** – Prețul per cameră, exprimat în euro (convertit din rupi indieni).
- **ReviewsCount** – Numărul total de recenzii primite de fiecare hotel.
- **Rating** – Nota medie acordată de utilizatori, indicând calitatea serviciilor.
- **City** – Orașul în care este localizat hotelul (ex: Scheveningen, Centrum, City Centre etc.).
- **State** – Provincia/regiunea în care se află hotelul (ex: South Holland, Rotterdam, Eindhoven, North Holland etc.).

ID	Name	Place	Type	Price	ReviewsCount	Rating	City	State
0	BUNK Hotel Amsterdam Amsterdam Noord, Am Bunk Pod for 2	Amsterdam Noord, Amsterdam	Am Bunk Pod for 2	86.757	778	8.4	Amsterdam Noord	Amsterdam
1	YOTEL Amsterdam	Amsterdam Noord, Amsterdam	Am Premium Double Room	167.937	500	8.1	Amsterdam Noord	Amsterdam
2	Motel One Amsterdam City Centre Double Room	Amsterdam Noord, Amsterdam	Double Room	143.693	1605	7.9	Amsterdam City Centre	Amsterdam
3	Innside Amsterdam Rai	Zuidas, Amsterdam	Innside Inow Double or Twin I	141.394	500	7.9	Zuidas	Amsterdam
4	Motel One Amsterdam Zuidasamstel, Amsterdam Double Room	Zuidas, Amsterdam	Double Room	104.581	500	8.8	Zuidas	Amsterdam
5	Innside by Melia Amst Zuidasamstel, Amsterdam The Innside Guestroom	Zuidas, Amsterdam	The Innside Guestroom	155.353	1264	8.4	Zuidas	Amsterdam
6	Eden Hotel Amsterdam Amsterdam City Centre Small Double Room	Amsterdam Noord, Amsterdam	Small Double Room	220.66	500	8.3	Amsterdam City Centre	Amsterdam
7	citizenM Amsterdam S Zuidasamstel, Amsterdam King Room	Zuidas, Amsterdam	King Room	156.266	500	8.8	Zuidas	Amsterdam
8	The Art Hotel Amsterdam City Centre Superior Double Room	Amsterdam Noord, Amsterdam	Superior Double Room	139.205	2069	7.7	Oud Zuid	Amsterdam
9	Aqua Amsterdam Prins Amsterdam City Centre Observatory King	Prins Hendrikplein, Amsterdam	King Room	409.419	500	8.9	Amsterdam City Centre	Amsterdam
10	Hyatt Regency Amsterd Amsterdam City Centre Twin Room	Amsterdam Noord, Amsterdam	Twin Room	343.508	500	8.5	Amsterdam City Centre	Amsterdam
11	NH Collection Amsterdam Amsterdam City Centre Superior Double or Twi	Amsterdam Noord, Amsterdam	Superior Double or Twin Room	323.301	500	8.6	Amsterdam City Centre	Amsterdam
12	Ibis Styles Amsterdam Amsterdam City Centre Small Double Room	Amsterdam Noord, Amsterdam	Small Double Room	195.778	500	7.9	Amsterdam City Centre	Amsterdam
13	Leonardo Royal Hotel Oude Amsterdam	Amsterdam Noord, Amsterdam	Queen Room - Disabili	124.696	500	8.5	Oost	Amsterdam
14	Residence Inn by Marriott Amsterdam City Centre	Amsterdam Noord, Amsterdam	Queen Room - Disabili	206.558	2974	8.9	Amsterdam City Centre	Amsterdam
15	Lemonade Boutique Hotel Zuid, Amsterdam	Amsterdam	Double or Twin Room	188.927	500	7.9	Oud Zuid	Amsterdam
16	Hotel V Neptunus	Amsterdam Noord, Amsterdam	Comfort Double or Twi	241.087	1278	8.9	Amsterdam City Centre	Amsterdam
17	Kimpton De Witt Amsterdam City Centre 1 Queen or King Bed Ei	Amsterdam Noord, Amsterdam	1 Queen or King Bed Ei	380.787	500	8.8	Amsterdam City Centre	Amsterdam
18	Qbic Hotel WTC Amst Zuidasamstel, Amsterdam Standard Double Room	Zuidas, Amsterdam	Standard Double Room	88.913	500	7.9	Zuidas	Amsterdam
19	Hotel Espresso	Oud West, Amsterdam	Small Double Room	142.912	1935	7.8	Oud West	Amsterdam
20	Hotel Central Amsterdam Oud Zuid, Amsterdam	Amsterdam	Double Room	74.734	518	6.7	Oud Zuid	Amsterdam
21	The Tulip Hotel Amsterdam City Centre Train Bank w/	Amsterdam Noord, Amsterdam	Train Bank w/	134.700	2084	7.8	Amsterdam City Centre	Amsterdam
22	Spiritsof Hotel Amsterdam	Amsterdam	Classic Double Room	233.145	1629	8.6	Amsterdam City Centre	Amsterdam
23	Holiday Inn Express Ar Amsterdam City Centre Double Room	Amsterdam Noord, Amsterdam	Double Room	190.388	500	8.4	Amsterdam City Centre	Amsterdam
24	Hotel Atlantis Amsterdam Oud Zuid, Amsterdam	Amsterdam	Twin Room	101.519	1783	7.4	Oud Zuid	Amsterdam
25	Home Green B Bemelen	Bemelen	Holiday Home	230.802	1	10	Bemelen	
26	Holiday Home Green B Bemelen	Bemelen	Holiday Home	291.874	1	9.6	Bemelen	

Această bază de date ne oferă o perspectivă detaliată asupra prețurilor, popularității și ratingurilor hotelurilor din diferite zone ale Olandei, permitându-ne să realizăm analize comparative și să identificăm modele de comportament ale consumatorilor.

PARTEA 1 – PROGRAMARE PYTHON

1. Liste

❖ *Descrierea problemei*

Problema abordată vizează administrarea și actualizarea ofertelor de destinații turistice, un aspect esențial pentru o platformă de rezervări, influențând atât experiența utilizatorilor, cât și poziționarea competitivă pe piață. În acest context, lista reprezintă o structură de date flexibilă, care permite adăugarea, eliminarea și modificarea elementelor, fiind ideală pentru gestionarea unui set dinamic de informații.

În această analiză vom utiliza o structură de tip listă pentru a gestiona destinațiile de vacanță și vom realiza următoarele operațiuni:

- Crearea și afișarea listei inițiale cu destinații turistice;
- Determinarea numărului de elemente din listă;
- Adăugarea și inserarea unor noi destinații;
- Stergerea unor elemente din listă;
- Inversarea ordinii destinațiilor;
- Golirea completă a listei;

❖ *Informatii necesare pentru rezolvare*

În cadrul analizei, am utilizat o lista cu destinații: Paris, Bucuresti, Maldive, Roma.

❖ *Produs software/functie/metode de calcul folosite*

- **Produs software folosit:** PyCharm
- **Limbaj:** Python
- **Functii/Metode utilizate:** `print()`, `len()`, `append()`, `insert()`, `pop()`, `reverse()`, `clear()`

❖ *Rezolvarea cu ajutorul produsului software*

Am creat o lista de destinații de vacanță, disponibile pe Booking.com, am afișat-o, utilizând funcția `print()`, ulterior am determinat lungimea listei, utilizând `len()`. Dupa care am extins lista, adăugând la finalul acesteia o nouă destinație, utilizând `append()`, și respectiv o nouă listă de destinații pe o pozitie specifică, utilizând `insert()`. Apoi, am sters lista de destinații adăugată anterior, am inversat elementele din lista utilizând `reverse()`, iar în final am golit lista, utilizând `clear()`.

CODUL UTILIZAT:

```
#Liste
#Crearea unei liste cu destinatii de vacanta
destinatii = ['Paris', 'Bucuresti', 'Maldivi', 'Roma']
#afisarea listei cu destinatii
print("Lista destinatiilor este:")
print(destinatii)
#Lungimea listei
print("Lungimea listei este:")
print(len(destinatii))
#Adaugarea la finalul listei a unei noi destinatii
destinatii.append('Amsterdam')
print("Lista dupa adaugarea noii destinatii:")
print(destinatii)
#Adaugarea unor noi destinatii pe pozitia 3
destinatii.insert(3, 'Berlin', 'Egipt', 'Londra'])
print("Lista dupa adaugarea unor noi destinatii pe pozitia 3:")
print(destinatii)
#Stergerea destinatiei de pe ultima pozitie
destinatii.pop()
print("Lista dupa stergerea ultimei destinatii:")
print(destinatii)
#Stergerea destinatiei de pe pozitia 3
destinatii.pop(3)
print("Lista dupa stergerea destinatiei de pe pozitia 3:")
print(destinatii)
#Inversarea elementelor din lista
destinatii.reverse()
print("Lista de destinatii inversata este:")
print(destinatii)
#Golirea listei
destinatii.clear()
print("Lista a fost golita!")
print(destinatii)
```

REZULTATELE OBTINUTE:

```
Lista destinatiilor este:
['Paris', 'Bucuresti', 'Maldivi', 'Roma']
Lungimea listei este:
4
Lista dupa adaugarea noii destinatii:
['Paris', 'Bucuresti', 'Maldivi', 'Roma', 'Amsterdam']
Lista dupa adaugarea unor noi destinatii pe pozitia 3:
['Paris', 'Bucuresti', 'Maldivi', ['Berlin', 'Egipt', 'Londra'], 'Roma', 'Amsterdam']
Lista dupa stergerea ultimei destinatii:
['Paris', 'Bucuresti', 'Maldivi', ['Berlin', 'Egipt', 'Londra'], 'Roma']
Lista dupa stergerea destinatiei de pe pozitia 3:
['Paris', 'Bucuresti', 'Maldivi', 'Roma']
Lista de destinatii inversata este:
['Roma', 'Maldivi', 'Bucuresti', 'Paris']
Lista a fost golita!
[]

Process finished with exit code 0
```

❖ Interpretarea rezultatelor

Rezultatele obținute demonstrează eficiența utilizării unei structuri de tip listă pentru gestionarea dinamică a ofertelor turistice. Toate operațiunile au fost realizate cu succes, iar lista a fost manipulată în conformitate cu obiectivele propuse. Acest tip de abordare permite actualizări rapide și eficiente ale destinațiilor turistice, oferind utilizatorilor o experiență îmbunătățită pe platforma de rezervări.

2. Tupluri

❖ Descrierea problemei

În analiza destinațiilor turistice, unul dintre factorii importanți care influențează decizia de călătorie este disponibilitatea unităților de cazare, în special a hotelurilor. Fiecare destinație turistică oferă un număr diferit de hoteluri, ceea ce poate afecta preferințele turiștilor. Prin urmare, este util să gestionăm aceste date într-o structură care asigură integritatea informațiilor, precum un tuplu.

Tuplurile sunt structuri de date similare listelor, dar nu pot fi modificate după crearea lor. Această caracteristică le face potrivite pentru stocarea unor date statice, cum ar fi numărul de hoteluri dintr-o destinație, deoarece aceste valori nu se schimbă frecvent.

Asadar, vom realiza urmatoarele operațiuni:

- Crearea și utilizarea unui tuplu pentru a gestiona numărul de hoteluri disponibile în patru destinații turistice;
- Accesarea și afișarea unui element specific din tuplu;
- Crearea unei funcții care calculează media numărului de hoteluri din aceste destinații;

❖ ***Informatii necesare pentru rezolvare***

Pentru a realiza această analiză, avem nevoie de următoarele informații:

- Lista destinațiilor turistice (utilizate anterior): Paris, București, Maldive, Roma.
- Numărul de hoteluri disponibile în fiecare dintre aceste destinații, conform Booking.com: Paris - 1792 hoteluri, București - 281 hoteluri, Maldive - 217 hoteluri, Roma - 993 hoteluri.

❖ ***Produs software/functii/metoda de calcul folosita***

- **Produs software folosit:** PyCharm
- **Limbaj:** Python
- **Functii/Metode utilizate:**
 - *print()*
 - Am definit o *funcție* în Python – *medie()* - pentru a calcula numărul mediu de hoteluri din cele 4 destinații turistice analizate. Funcția primește ca parametri numărul de hoteluri din fiecare destinație și returnează atât valorile inițiale, cât și media acestora.

❖ ***Rezolvarea cu ajutorul produsului software***

Am creat un tuplu care conține numărul de hoteluri disponibile pentru fiecare dintre cele patru destinații turistice analizate. Apoi, am accesat și afișat numărul de hoteluri de pe poziția a treia din tuplu. Pentru a calcula media hotelurilor din cele patru destinații, am definit o funcție numită *medie()*, care primește patru valori corespunzătoare numărului de hoteluri și returnează atât valorile inițiale, cât și media acestora. Am apelat funcția și am salvat rezultatul într-o variabilă, după care am extras și afișat separat valoarea mediei.

CODUL UTILIZAT:

```
#Tupluri
#Crearea unui tuplu cu nr de hoteluri pt fiecare destinatie
nr_hoteluri = (1792, 281, 217, 993)
#Afisarea numarului de hoteluri de pe pozitia 3
print(nr_hoteluri[3])

#Calcularea numarului mediu de hoteluri din cele 4 destinatii
def medie(a,b,c,d):
    return a,b,c,d, (a+b+c+d)/4

rezultat = medie( a=1792, b=281, c=217, d=993)
print(rezultat)
medie = rezultat[4]
print(medie)
```

REZULTATELE OBTINUTE:

```
Numarul de hoteluri pentru destinatia de pe pozitia 3 este:
993
Rezultatul obtinut in urma apelarii functiei:
(1792, 281, 217, 993, 820.75)
Numarul mediu de hoteluri din cele 4 destinatii este:
820.75
```

❖ *Interpretarea rezultatelor*

Rezultatele obținute oferă o perspectivă asupra distribuției numărului de hoteluri în cele patru destinații analizate. În primul rând, prin accesarea poziției a treia din tuplu, am obținut 993, ceea ce indică faptul că această destinație are un număr considerabil de hoteluri disponibile.

Apoi, apelând funcția `medie()`, am obținut un tuplu care conține atât valorile inițiale ale numărului de hoteluri pentru fiecare destinație, cât și media calculată: (1792, 281, 217, 993, 820.75). În final, extragerea și afișarea mediei ne arată că numărul mediu de hoteluri disponibile în cele patru destinații analizate este 820.75. Acest rezultat sugerează o variație semnificativă între destinații, existând locații cu un număr mult mai mare de hoteluri, ceea ce poate indica o popularitate mai ridicată sau o ofertă turistică mai dezvoltată.

3. Dictionare

❖ *Descrierea problemei*

În cadrul acestei secțiuni, gestionăm informații despre destinațiile de vacanță și numărul de hoteluri disponibile în fiecare locație folosind un dicționar în Python.

Dicționarele sunt structuri de date care permit stocarea perechilor cheie-valoare, oferind un mod eficient de a accesa și manipula datele prin intermediul cheilor unice.

Scopul analizei este de a demonstra utilizarea dicționarelor pentru organizarea datelor într-un format clar și structurat, precum și efectuarea unor operațiuni de accesare, actualizare și afișare a informațiilor relevante despre destinațiile de vacanță.

Operatiunile realizate sunt urmatoarele:

- Crearea unui dicționar;
- Accesarea unei valori folosind cheia;
- Adăugarea unei noi destinații;
- Accesarea unei valori;
- Actualizarea unei valori existente;
- Afisarea tuturor elementelor dicționarului;
- Parcursarea dicționarului ,utilizand structura repetitiva for;

❖ *Informatii necesare pentru rezolvare*

Pentru a rezolva această problemă, avem nevoie de următoarele informații:

- Lista destinațiilor de vacanță și numărul de hoteluri disponibile în fiecare locație, conform Booking.com.

❖ *Produs software/functii/metode de calcul folosite*

- **Produs software folosit:** PyCharm
- **Limbaj:** Python
- **Functii/Metode utilizate:** `print()`, `structuri repetitive for`, `dict.get(cheie)`, `dict.items()`, `dict.keys()`, `dict.values()`.

❖ *Rezolvarea cu ajutorul produsului software*

Am creat un dicționar care conține numărul de hoteluri disponibile pentru fiecare dintre destinațiile turistice analizate. Pentru a accesa și afișa numărul de hoteluri dintr-o anumită destinație, am utilizat cheia corespunzătoare în dicționar. Am adăugat o nouă destinație împreună cu numărul său de hoteluri și am folosit metoda `get()` pentru a afișa valoarea asociată acestei noi destinații. Ulterior, am actualizat numărul de hoteluri pentru o destinație existentă, modificând valoarea corespunzătoare cheii. Pentru a afișa toate elementele dicționarului, am utilizat metoda `items()`, iterând prin fiecare pereche cheie-valoare și afișând informațiile într-un format structurat. În final, am utilizat metodele `keys()` și `values()` pentru a extrage și afișa separat listele de chei și valori din dicționar.

CODUL UTILIZAT:

```
#Dicționar
#Părarea unui dicționar cu destinații și nr de hoteluri
dicționar = {"Paris":1792,
             "Bucuresti":281,
             "Maldivi":217,
             "Roma":993
            }
print(dicționar)
#afișarea numărului de hoteluri pentru o destinație, folosind cheia
print("Numărul de hoteluri din Bucuresti este:")
print(dicționar["Bucuresti"])
#Adaugarea unei noi destinații
dicționar["Amsterdam"] = 476
print(dicționar)
#afișarea numărului de hoteluri pentru destinația "Amsterdam"
print("Numărul de hoteluri din Amsterdam este:")
print(dicționar.get("Amsterdam"))
#Actualizarea numărului de hoteluri pentru destinația "Bucuresti" la 300
dicționar["Bucuresti"] = 300
print(dicționar)

print(dicționar.items())
for cheie, valoare in dicționar.items():
    print(f"In destinația {cheie} există {valoare} hoteluri.")

print(dicționar.keys())
print(dicționar.values())
```

REZULTATELE OBTINUTE:

```
{'Paris': 1792, 'Bucuresti': 281, 'Maldivi': 217, 'Roma': 993}
Numărul de hoteluri din Bucuresti este:
281
{'Paris': 1792, 'Bucuresti': 281, 'Maldivi': 217, 'Roma': 993, 'Amsterdam': 476}
Numărul de hoteluri din Amsterdam este:
476
{'Paris': 1792, 'Bucuresti': 300, 'Maldivi': 217, 'Roma': 993, 'Amsterdam': 476}
dict.items([('Paris', 1792), ('Bucuresti', 300), ('Maldivi', 217), ('Roma', 993), ('Amsterdam', 476)])
In destinația Paris există 1792 hoteluri.
In destinația Bucuresti există 300 hoteluri.
In destinația Maldivi există 217 hoteluri.
In destinația Roma există 993 hoteluri.
In destinația Amsterdam există 476 hoteluri.
dict_keys(['Paris', 'Bucuresti', 'Maldivi', 'Roma', 'Amsterdam'])
dict_values([1792, 300, 217, 993, 476])
```

❖ Interpretarea rezultatelor

Dicționarul a fost creat pentru a stoca informații despre numărul de hoteluri disponibile în mai multe destinații turistice. Inițial, acesta conține patru orașe: Paris, București, Maldivi și Roma, împreună cu numărul corespunzător de hoteluri. Pentru a verifica accesibilitatea datelor, a fost afișat numărul de hoteluri din București, confirmând că acesta este 281. Ulterior, dicționarul

a fost extins prin adăugarea destinației Amsterdam, cu 476 hoteluri, iar această modificare a fost verificată prin extragerea numărului de hoteluri aferent orașului nou adăugat. În continuare, numărul de hoteluri din București a fost actualizat de la 281 la 300, demonstrând capacitatea dicționarului de a gestiona modificări ale datelor. Pentru o vizualizare clară a tuturor informațiilor stocate, s-au afișat atât perechile cheie-valoare, cât și listele separate ale destinațiilor și ale numărului de hoteluri. De asemenea, fiecare destinație a fost prezentată individual, printr-un mesaj personalizat care indică numărul de hoteluri disponibile. Aceste operațiuni confirmă flexibilitatea și eficiența dicționarului în stocarea, actualizarea și accesarea datelor relevante despre destinațiile turistice.

4. Seturi

❖ *Descrierea problemei*

În industria ospitalității, facilitățile oferite de hoteluri joacă un rol esențial în atragerea și satisfacția clienților. Turiștii caută hoteluri care dispun de anumite facilități precum WiFi, piscină, parcare sau spa, iar disponibilitatea acestor servicii influențează decizia de rezervare. De asemenea, rating-ul hotelurilor este un factor crucial în selecția căzării, iar turiștii preferă să comparați hotelurile pe baza acestuia. Prin urmare, este util să gestionăm aceste informații într-o structură eficientă pentru analiză și comparare.

Astfel, vom realiza următoarele operațiuni:

- Gestionarea facilităților hotelurilor folosind seturi pentru a determina facilitățile comune, distincte sau disponibile în totalitate;
- Crearea unei liste de tupluri pentru a stoca informații despre hotelurile dintr-o anumită destinație;
- Utilizarea operațiunilor `intersection()`, `union()` și `difference()` pentru a compara hotelurile și a identifica facilități comune între hoteluri, facilități distincte (ce oferă un hotel și nu celălalt), lista tuturor facilităților disponibile în cel puțin unul dintre hoteluri, cat și verificarea existenței unei facilități într-un anumit hotel;

❖ *Informatii necesare pentru rezolvare*

Pentru a realiza această analiză, avem nevoie de lista facilităților oferite de cele două hoteluri.

❖ *Produs software/functii/metoda de calcul folosita*

- **Produs software folosit:** PyCharm
- **Limbaj:** Python
- **Functii/Metode utilizate:**

- `print()` – pentru afișarea rezultatelor.
- `set()` – pentru definirea mulțimilor de facilități ale hotelurilor.
- `in` – pentru verificarea existenței unei facilități într-un hotel.
- `intersection() / &` – pentru a determina facilitățile comune.
- `union()` – pentru a combina toate facilitățile oferite de ambele hoteluri.
- `difference() / - -` – pentru a identifica facilitățile exclusive fiecărui hotel.

❖ *Rezolvarea cu ajutorul produsului software*

În cadrul implementării, s-au utilizat seturi pentru a stoca facilitățile oferite de două hoteluri. S-a demonstrat eliminarea duplielor prin utilizarea `set("Hoteluri de pe Booking")`. Ulterior, s-a verificat prezența unei facilități specifice în set prin expresia `'wifi'` în `hotel1`. Pentru compararea ofertelor celor două hoteluri, s-au aplicat mai multe operațiuni: intersecția (`hotel1.intersection(hotel2)` sau `hotel1 & hotel2`), reuniunea (`hotel1.union(hotel2)`) și diferența (`hotel1.difference(hotel2)`, respectiv `hotel2 - hotel1`).

CODUL UTILIZAT:

```
#Seturi
print(set("Hoteluri de pe Booking"))

hotel_1 = {'wifi', 'piscina', 'parcare'}
hotel_2 = {'wifi', 'piscina', 'spa'}

#Are hotelul 1 wifi?
print("Are hotelul 1 wifi?")
print('wifi' in hotel_1)

#Care sunt facilitatile comune din cele doua hoteluri?
print("Care sunt facilitatile comune din cele doua hoteluri?")
print(hotel_1.intersection(hotel_2))
print(hotel_1 & hotel_2)

#Sa se afiseze toate facilitatile
print("Afisarea tuturor facilitatilor disponibile:")
print(hotel_1.union(hotel_2))
print(hotel_1 | hotel_2)

#Ce facilitati are hotelul 1 si nu are hotelul 2?
print("Ce facilitati are hotelul 1 si nu are hotelul 2?")
print(hotel_1.difference(hotel_2))
#Ce facilitati are hotelul 2 si nu are hotelul 1?
print("Ce facilitati are hotelul 2 si nu are hotelul 1?")
print(hotel_2 - hotel_1)
```

REZULTATELE OBTINUTE:

```
{'o', 'u', 't', 'i', 'd', 'p', ' ', 'g', 'k', 'n', 'B', 'r', 'H', 'l'}
Are hotelul 1 wifi?
True
Care sunt facilitatile comune din cele doua hoteluri?
{'piscina', 'wifi'}
{'piscina', 'wifi'}
Afisarea tuturor facilitatilor disponibile:
{'parcare', 'piscina', 'spa', 'wifi'}
{'parcare', 'piscina', 'spa', 'wifi'}
Ce facilitati are hotelul 1 si nu are hotelul 2?
{'parcare'}
Ce facilitati are hotelul 2 si nu are hotelul 1?
{'spa'}
```

❖ Interpretarea rezultatelor

După aplicarea operațiunii `set(" Hoteluri de pe Booking")`, rezultatul a fost un set de caractere unice, fără duplicate și fără a păstra ordinea originală: `{'n', 'g', 'k', 'r', 'o', ' ', 'l', 'd', 'p', 'B', 't', 'u', 'i', 'H', 'e'}`. Verificarea existenței facilității „`wifi`” în `hotel1` a returnat `True`, confirmând că acest serviciu este disponibil. Intersecția facilităților oferite de `hotel1` și `hotel2` a rezultat în `{'piscina', 'wifi'}`, indicând că ambele hoteluri oferă aceste servicii. Reuniunea celor două seturi a generat `{'spa', 'parcare', 'piscina', 'wifi'}`, ceea ce înseamnă că, împreună, cele două hoteluri oferă toate aceste facilități. Diferența dintre `hotel1` și `hotel2` a returnat `{'parcare'}`, ceea ce arată că doar `hotel1` oferă această facilitate, în timp ce diferența inversă (`hotel2 - hotel1`) a avut ca rezultat `{'spa'}`, indicând că doar `hotel2` pune la dispoziție acest serviciu.

5. Definirea si apelarea functiilor, structuri conditionate si repetitive

❖ Descrierea problemei

Analiza hotelurilor dintr-o destinație turistică reprezintă un factor esențial pentru turiști, deoarece aspecte precum rating-ul hotelurilor sau disponibilitatea unor facilități pot influența decizia de cazare. Pentru a gestiona aceste informații eficient, am utilizat liste de tupluri pentru a reprezenta hotelurile din Paris, fiecare având un identificator unic, un nume, un rating și o indicație dacă acceptă sau nu animale de companie. Ulterior, am definit funcții pentru a analiza aceste date.

Astfel, vom realiza următoarele operațiuni:

- Crearea unei liste de tupluri care conține informațiile despre hotelurile selectate;
- Definirea unei funcții care calculează rating-ul mediu al hotelurilor;
- Implementarea unei funcții de sortare a hotelurilor în funcție de rating;
- Crearea unei funcții care verifică dacă un hotel acceptă animale de companie;

❖ Informatii necesare pentru rezolvare

Pentru a efectua analiza, avem nevoie de lista hotelurilor din Paris cu următoarele atribute: ID hotel, Denumire, Rating, Acceptare animale de companie.

❖ Produs software/functii/metoda de calcul folosita

- **Produs software folosit:** PyCharm
- **Limbaj:** Python
- **Functii/Metode utilizate:**

Am definit urmatoarele trei funcții pentru gestionarea informațiilor despre hoteluri:

- Funcția **rating_mediul(hoteluri)** ce calculează media ratingurilor pentru o listă de hoteluri. Aceasta primește ca parametru o listă de tupluri, fiecare conținând ID-ul hotelului, denumirea, ratingul și informația privind acceptarea animalelor de companie. Funcția parcurge lista, adună valorile ratingurilor și returnează media acestora.
- Funcția **sortare_hoteluri(lista)** sortează hotelurile în ordine descrescătoare în funcție de rating. Aceasta primește o listă de hoteluri și utilizează funcția **sorted()** cu un criteriu bazat pe rating (al treilea element din tuplu), setând **reverse=True** pentru a obține ordonarea descendenta.
- Funcția **accepta_animale(hoteluri, id_hotel)** verifică dacă un hotel specific acceptă animale de companie. Aceasta primește ca parametri lista de hoteluri și ID-ul unui hotel, iar apoi parcurge lista pentru a identifica hotelul cu ID-ul respectiv. Dacă hotelul este găsit, funcția returnează un mesaj care indică dacă acesta permite sau nu accesul animalelor de companie. În cazul în care ID-ul nu este găsit, se returnează un mesaj corespunzător.

❖ Rezolvarea cu ajutorului produsului software

Pentru a analiza hotelurile, am creat o listă de tupluri care conține informațiile relevante despre fiecare hotel. Am definit apoi o funcție **rating_mediul()** care parcurge lista hotelurilor, adună rating-urile și returnează media acestora. Apoi, am implementat funcția **sortare_hoteluri()**, care utilizează funcția **sorted()** cu un lambda pentru a sorta hotelurile descrescător după rating. De asemenea, am realizat o funcție **accepta_animale()**, care verifică dacă un hotel specific acceptă animale de companie, returnând un mesaj corespunzător.

Am apelat fiecare funcție și am afișat rezultatele obținute, facilitând astfel analiza și selecția hotelurilor în funcție de preferințele turiștilor.

CODUL UTILIZAT:

```
#Funcții
#Crearea unei liste de tupluri cu hoteluri din Paris (id, denumire, rating și dacă acceptă sau nu animalele de companie
hoteluri = [(1, 'Hôtel Le Montmartre Saint Pierre', 8.9, 'Nu'),
             (2, '1.75 Paris Le Charme', 9.2, 'Da'),
             (3, 'Hôtel Clémence', 8.7, 'Nu'),
             (4, 'St Honore Champs Elysees by Edelsam', 9.5, 'Da'),
             (5, 'Hôtel Jarry Confort', 6.7, 'Da')]

#Definirea unei funcții care să calculeze ratingul mediu
def rating_mediul(hoteluri):
    suma = 0
    for hotel in hoteluri:
        suma = suma + hotel[2]
    nr_hoteluri = len(hoteluri)
    return suma/nr_hoteluri

medie_ratinguri = rating_mediul(hoteluri)
print(f"Media ratingurilor este:{medie_ratinguri}")

#Definirea unei funcții care să sorteze descendent hotelurile în funcție de rating

def sortare_hoteluri(lista):
    var_liste = []
    for i in lista:
        var_liste = sorted(lista, key = lambda hotel: hotel[2], reverse = True)
    return var_liste

sort_hoteluri = sortare_hoteluri(hoteluri)
print(f"Sortarea hotelurilor după rating: {sort_hoteluri}")

#Crearea unei funcții care să verifice dacă hotelul acceptă sau nu animale
def accepta_animale(hoteluri, id_hotel):
    for hotel in hoteluri:
        if hotel[0] == id_hotel:
            return f"Hotelul {hotel[1]} acceptă animale de companie." if hotel[3] == 'Da' else f"Hotelul {hotel[1]} "
            f"NU acceptă animale de companie."
    return "Hotelul nu a fost găsit în listă."

print(accepta_animale(hoteluri, id_hotel[1]))
print(accepta_animale(hoteluri, id_hotel[2]))
```

REZULTATELE OBTINUTE:

```
Media ratingurilor este:8.6
Sortarea hotelurilor după rating: [(4, 'St Honore Champs Elysees by Edelsam', 9.5, 'Da'), (2, '1.75 Paris Le Charme', 9.2, 'Da'), (1, 'Hôtel Le Montmartre Saint Pierre', 8.9, 'Nu'), (3, 'Hôtel Clémence', 8.7, 'Nu'), (5, 'Hôtel Jarry Confort', 6.7, 'Da')]
Hotelul Hôtel Le Montmartre Saint Pierre NU acceptă animale de companie.
Hotelul 1.75 Paris Le Charme acceptă animale de companie.
```

❖ Interpretarea rezultatelor

Rezultatele obținute reflectă analiza celor 5 hoteluri din Paris. Media rating-ului acestora este de 8.6, ceea ce indică un nivel general ridicat de satisfacție din partea clienților. Hotelurile au fost sortate în ordine descrescătoare a rating-ului, iar rezultatul arată că St Honore Champs Elysees by Edelsam este cel mai apreciat cu un rating de 9.5, urmat de 1.75 Paris Le Charme cu 9.2. Alte hoteluri, precum Hôtel Le Montmartre Saint Pierre și Hôtel Clémence, au rating-uri bune, de 8.9 și respectiv 8.7, în timp ce Hôtel Jarry Confort, cu un rating de 6.7, este cel mai slab cotat. În ceea ce privește politica de acceptare a animalelor de companie, Hôtel Le Montmartre Saint Pierre nu acceptă animale, în timp ce 1.75 Paris Le Charme permite acest lucru. Aceste informații sunt esențiale pentru turiștii care doresc să călătoresc cu animalele lor de companie.

6. Importul unui fisier CSV

❖ Descrierea problemei

Pentru analiza dataset-ului utilizat în proiect, am realizat o prima explorare a datelor. Aceasta include încărcarea fișierului CSV, vizualizarea conținutului tabelului, verificarea dimensiunii acestuia, afișarea numelor coloanelor și identificarea tipurilor de date conținute. Aceste operațiuni ajută la înțelegerea structurii dataset-ului și la pregătirea acestuia pentru prelucrare ulterioară.

❖ *Informatii necesare pentru rezolvare*

Pentru a realiza această analiză, am utilizat un fișier CSV denumit HotelDataset.csv, care conține informații despre hoteluri. Acest fișier a fost încărcat în Pandas DataFrame folosind codificarea ISO-8859-1 pentru a evita probleme legate de caractere speciale.

❖ *Produs software/functii/metoda de calcul folosita*

- **Produs software folosit:** PyCharm
- **Limbaj:** Python
- **Functii/Metode utilizate:**
 - Librăria Pandas pentru manipularea și explorarea datelor.
 - Funcția `pd.read_csv()` pentru citirea fișierului CSV și stocarea acestuia într-un DataFrame.
 - Metodele `.head()`, `.columns`, `.info()`, `.shape` pentru obținerea unor informații esențiale despre dataset: `.head()` – afișează primele 5 rânduri, `.columns` – returnează numele coloanelor din dataset, `.info()` – oferă detalii despre tipurile de date și valorile lipsă, `.shape` – returnează un tuplu cu numărul de rânduri și coloane.

❖ *Rezolvarea cu ajutorul produsului software*

In continuare, am importat un fișier CSV utilizând funcția `read_csv` din biblioteca Pandas. Fișierul este citit de la locația specificată pe disc și se folosește encoding-ul ISO-8859-1, deoarece fișierul nu este salvat în formatul implicit UTF-8. După ce fișierul este încărcat în dataframe-ul Pandas (df), se afișează setul de date complet folosind `print(df)`, urmat de vizualizarea primelor 5 rânduri din dataset cu `df.head()`. Se obțin și informații suplimentare despre coloanele dataset-ului prin utilizarea `df.columns`, care listează numele acestora, iar cu `df.info()`, sunt prezentate detalii despre tipul de date și numărul de valori nenule din fiecare coloană. În final, `df.shape` afișează un tuplu care indică dimensiunea dataset-ului, respectiv numărul de rânduri și coloane.

CODUL UTILIZAT:

```
#Importarea fisierului csv
#sursa date: https://www.kaggle.com/datasets/mukuldeshantri/hotels-netherlands?resource=download

df = pd.read_csv( filepath_or_buffer: r"0:\PyCharm_Projects\.venv\HotelDataset.csv", encoding="ISO-8859-1")

#Afisarea setului de date
print(df)
#Afisarea primelor 5 randuri din dataset
print(df.head())
#Afisarea coloanelor
print(df.columns)
#Afiseaza informatii despre coloane
print(df.info())
#Afiseaza un tuplu cu nr de linii si nr de coloane
print(df.shape)
```

REZULTATELE OBTINUTE:

ID	...	State
0	0	Amsterdam
1	1	Amsterdam
2	2	Amsterdam
3	3	Amsterdam
4	4	Amsterdam
...
520	520	NaN
521	521	NaN
522	522	NaN
523	523	NaN
524	524	NaN

[525 rows x 9 columns]

ID	Name	...	City	State
0	BUNK Hotel	Amsterdam	Amsterdam Noord	Amsterdam
1	YOTEL Amsterdam	...	Amsterdam Noord	Amsterdam
2	Multatului Hotel	...	Amsterdam City Center	Amsterdam
3	nhow Amsterdam Rai	...	Zuidermanstel	Amsterdam
4	Motel One Amsterdam	...	Zuidermanstel	Amsterdam

[5 rows x 9 columns]

```
Index(['ID', 'Name', 'Place', 'Type', 'Price', 'ReviewsCount', 'Rating',
       'City', 'State'],
      dtype='object')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525 entries, 0 to 524
```

#	Column	Non-Null Count	Dtype
0	ID	525 non-null	int64
1	Name	525 non-null	object
2	Place	525 non-null	object
3	Type	525 non-null	object
4	Price	525 non-null	float64
5	ReviewsCount	512 non-null	float64
6	Rating	515 non-null	float64
7	City	525 non-null	object
8	State	133 non-null	object

dtypes: float64(3), int64(1), object(5)
memory usage: 37.0+ KB
None
(525, 9)

❖ Interpretarea rezultatelor

In urma analizarii rezultatelor, observam ca setul de date conține 525 de rânduri și 9 coloane, reprezentând informații despre hoteluri din Olanda. Coloanele sunt reprezentate de ID-ul, numele, locația, tipul, prețul, numărul de recenzii, ratingul, orașul și statul hotelurilor, toate aparținând orașului Amsterdam. Observam ca valorile lipsă au fost notate cu NaN. În ceea ce privește valorile nenele, coloanele ReviewsCount și Rating au câteva valori lipsă (13, respectiv 10), iar coloana State prezintă cele mai multe valori lipsă, având doar 133 de valori complete din 525 de înregistrări. Coloanele ID, Name, Place, Type, Price, și City nu conțin valori lipsă, iar tipurile de date sunt corect alocate, cu valori numerice pentru ID, ReviewsCount, Rating, și Price, și object/date categoriale pentru celelalte coloane. In final, a fost afisat un tuplu cu numarul de inregistrari si numarul de coloane, utilizand shape().

7. Tratarea valorilor lipsă. Stergerea coloanelor si a inregistrarilor.

❖ Descrierea problemei

Setul de date importat conține valori lipsă în coloanele ReviewsCount (13 valori lipsă), Rating (10 valori lipsă) și State (392 valori lipsă). Se dorește tratarea acestor valori lipsă pentru a asigura o analiză coerentă și relevantă. În acest sens, valorile lipsă din ReviewsCount și Rating vor fi completate utilizând media grupată, calculată pe baza unor caracteristici relevante, cum ar fi orașul (City). Pentru coloana State, valorile lipsă au fost completate căutând informațiile corespunzătoare online, pentru a asigura corectitudinea acestora. De asemenea, se dorește eliminarea coloanei Place, deoarece conține informații redundante, precum orașul și statul, care sunt deja prezente în alte coloane.

❖ Informatii necesare pentru rezolvare

Pentru realizarea acestor operații avem nevoie de datele din fisierul CSV.

❖ Produs software/functii/metoda de calcul folosita

- **Produs software folosit:** PyCharm
- **Limbaj:** Python
- **Functii/Metode utilizate:**
 - *isna()* - Verifică dacă există valori lipsă (NaN) în cadrul setului de date.
 - *sum()* - Calculează suma valorilor NaN pe fiecare coloană.
 - *groupby()* - Grupează datele pe baza unei anumite coloane (în acest caz, City), pentru a aplica o funcție pe fiecare grup.
 - *transform()* - Permite aplicarea unei funcții de transformare pe fiecare grup de date.
 - *fillna()* - Completează valorile lipsă cu o valoare specificată, în acest caz, media grupului.
 - *unique()* - Returnează valorile unice dintr-o coloană.
 - *apply()* - Aplica o funcție pe fiecare rând sau coloană a unui DataFrame, în acest caz pentru completarea valorilor din coloana State.
 - *get()* - Aplica o funcție pe fiecare rând sau coloană a unui DataFrame, în acest caz pentru completarea valorilor din coloana State.
 - *drop()* - Șterge o coloană sau un rând din DataFrame (în acest caz, coloana Place).
 - *to_csv()* - Salvează DataFrame-ul într-un fișier CSV.

❖ *Rezolvarea cu ajutorul produsului software*

Am început prin verificarea valorilor lipsă din setul de date utilizând metoda `df.isna()`, care a returnat un DataFrame ce indică locurile unde există valori NaN, și am calculat numărul de valori lipsă pe fiecare coloană cu `df.isna().sum()`. Am observat că în coloanele ReviewsCount și Rating existau valori lipsă, astfel că am decis să le trătam folosind media grupată pe baza coloanei City, aplicând `groupby('City')` și `transform(lambda x: x.fillna(x.mean()))` pentru a înlocui valorile lipsă cu media grupului corespunzător. După această transformare, am verificat că nu mai există valori lipsă în aceste coloane. În ceea ce privește coloana State, am constatat că aveam un număr semnificativ de valori lipsă (392), așa că am ales să le completam manual, utilizând un dicționar care asociază orașele cu statele corespunzătoare. Am folosit metoda `apply()` pentru a aplica completările în coloana State. După completarea manuală a valorilor lipsă, am verificat dacă mai există valori lipsă în această coloană. Ulterior, am eliminat coloana Place, deoarece conținea informații redundante, având deja City și State în setul de date. În final, am salvat setul de date curățat într-un fișier CSV numit "HotelDataset_final.csv", fără a include indexul.

CODUL UTILIZAT:

```
#Verificam daca avem valori lipsa
print(df.isna())
print("Numarul de valori lipsa per coloana:")
print(df.isna().sum())

#Observam ca in coloanele 'ReviewsCount' si 'Rating' avem valori lipsa (13, respectiv 10 valori lipsa)
#Vom trata valorile lipsa utilizand media grupata
df['ReviewsCount'] = df.groupby('City')['ReviewsCount'].transform(lambda x: x.fillna(x.mean()))
df['Rating'] = df.groupby('City')['Rating'].transform(lambda x: x.fillna(x.mean()))

print("Afisarea valorilor lipsa:")
print(df.isna().sum())
#Observam ca nu mai exista valori lipsa in cele doua coloane

#Observam ca in cadrul coloanei 'State' avem foarte multe valori lipsa (392 valori lipsa), iar in continuare
# le vom completa manual pentru a nu fi nevoit sa stergem inregistrarile

#Completarea coloanei 'State' acolo unde avem valori lipsa
#Identificarea oraselor care nu au statul completat
missing_states = df[df["State"].isna()]["City"].unique()
```

```
#Afisarea oraselor respective
print("Orasele care necesita completarea statului:")
print(missing_states)

#Completarea manuala a valorilor pentru statele lipsa
manual_entries = {
    "Bemelen": "Limburg",
    "Zwolle": "Overijssel",
    "Breda": "North Brabant",
    "Delft": "South Holland",
    "Den Bosch": "North Brabant",
    "Groningen": "Groningen",
    "Haarlem": "North Holland",
    "The Hague": "The Hague",
    "Hoofddorp": "North Holland",
    "Leeuwarden": "Friesland",
    "Maastricht": "Maastricht",
    "Middelburg": "Zeeland",
    "Nijmegen": "Gelderland",
    "Roermond": "Limburg",
    "Scheveningen": "The Hague",
    "Utrecht": "Utrecht",
    "Valkenburg": "Limburg",
    "Vlissingen": "Zeeland",
    "Voorthuizen": "Gelderland",
    "Zandvoort": "North Holland",
    ...
```

```

        "Zwolle": "Overijssel",
        "Eindhoven": "Eindhoven"
    }

#Completarea valorilor lipsă în 'State'
df["State"] = df.apply(lambda row: manual_entries.get(row["City"], row["State"]), axis=1)

#Verificam daca mai avem valori NaN in 'State'
print("Afisarea valorilor lipsa din coloana State:")
print(df[df["State"].isna()])

#Stergerea coloanei 'Place', deoarece este redundanta
df = df.drop( labels='Place', axis = 1)
print("Coloanele din setul de date:")
print(df.columns)

#Salvam rezultatul într-un fișier CSV
df.to_csv( path_or_buf= "HotelDataset_final.csv", index=False)

```

REZULTATELE OBTINUTE:

ID	Name	Place	Type	Price	ReviewsCount	Rating	City	State
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...
520	False	False	False	False	False	False	False	True
521	False	False	False	False	False	False	False	True
522	False	False	False	False	False	False	False	True
523	False	False	False	False	False	False	False	True
524	False	False	False	False	False	False	False	True

[525 rows x 9 columns]

Numarul de valori lipsă per coloana:

ID	Name	Place	Type	Price	ReviewsCount	Rating	City	State
0	0	0	0	0	13	10	0	392
ReviewsCount								
Rating								
City								
State								
dtype: int64								

Afisarea valorilor lipsă:

```

ID          0
Name         0
Place        0
Type          0
Price         0
ReviewsCount  0
Rating        0
City          0
State       392
dtype: int64

```

Orașele care necesită completarea statului:

```

['Bemelen' 'Breda' 'Delft' 'Den Bosch' 'Eindhoven' 'Groningen' 'Haarlem'
 'The Hague' 'Hoofddorp' 'Leeuwarden' 'Maastricht' 'Middelburg' 'Nijmegen'
 'Roermond' 'Scheveningen' 'Utrecht' 'Valkenburg' 'Vlissingen'
 'Voorthuizen' 'Zandvoort' 'Zwolle']

```

Afisarea valorilor lipsă din coloana State:

```

Empty DataFrame
Columns: [ID, Name, Place, Type, Price, ReviewsCount, Rating, City, State]
Index: []

```

Coloanele din setul de date:

```

Index(['ID', 'Name', 'Type', 'Price', 'ReviewsCount', 'Rating', 'City',
       'State'],
      dtype='object')

```

❖ Interpretarea rezultatelor

Rezultatele arată că doar coloanele ReviewsCount, Rating și State conțin valori lipsă. Am tratat valorile lipsă din ReviewsCount și Rating utilizând media grupată pe orașe, iar acum aceste coloane nu mai au valori lipsă. În cazul coloanei State, am identificat 21 de orașe ce necesită completarea manuală a valorilor lipsă. De asemenea, am eliminat coloana redundantă Place. În urma tratării valorilor lipsă, setul de date este pregătit pentru analiza ulterioară.

8. Analiza datelor din fisierul CSV. Analiza descriptiva.

❖ Descrierea problemei

În cadrul proiectului, analiza descriptivă a datelor este un pas esențial pentru a obține o înțelegere de bază a setului de date. Obiectivul principal este să identificăm caracteristicile și tipurile generale din datele referitoare la hoteluri, cum ar fi prețurile, numărul de recenzii și ratingurile, pentru a sprijini analizele ulterioare.

❖ *Informatii necesare pentru rezolvare*

Pentru această etapă de analiză, am utilizat un fișier CSV care conține informații despre hoteluri, precum: Nume, Tip, Preț, Număr recenzii, Rating, Oraș, și Stat. Aceste date sunt esențiale pentru a obține statistică descriptivă, cum ar fi valorile minime, maxime, medii și valorile unice din fiecare coloană.

❖ *Produs software/functii/metoda de calcul folosită*

- **Produs software folosit:** PyCharm
- **Limbaj:** Python
- **Functii/Metode utilizate:**
 - `df.describe()`: Calculează statistică descriptivă pentru coloanele numerice.
 - `df.describe(include='object')`: Calculează statistică descriptivă pentru coloanele de tip string.
 - `df.select_dtypes(include=[np.number])`: Selectează doar coloanele numerice pentru analize suplimentare.
 - `df.head(), df.tail()`: Afisează primele și ultimele valori dintr-o coloană.
 - `df.nunique()`: Afisează numărul de valori unice din fiecare coloană.
 - `df[column].unique()`: Returnează valorile unice dintr-o coloană.
 - `df[column].min(), df[column].max(), df[column].mean(), df[column].median()`: Calculează valorile minime, maxime, medii și mediane pentru o coloană numerică.
 - `df[column].idxmax()`: Returnează indexul valorii maxime dintr-o coloană.

❖ *Rezolvarea cu ajutorului produsului software*

Am citit fișierul CSV folosind `pd.read_csv()` și am utilizat funcțiile `df.head()`, `df.tail()`, și `df.columns` pentru a vizualiza setul de date, primele și ultimele valori ale coloanei „Name” și numele tuturor coloanelor. Am folosit `df.describe()` pentru a obține statistică generală pentru coloanele numerice, cum ar fi minimele, maximele, medii și deviațiile standard. De asemenea, pentru coloanele de tip text, am utilizat `df.describe(include='object')` pentru a obține statistică precum numărul de valori unice și cele mai frecvente valori. Am identificat coloanele numerice folosind `df.select_dtypes(include=[np.number])`, iar apoi am folosit `df.Price.describe()` pentru a analiza prețurile hotelurilor și pentru a extrage statistică relevantă. Pentru a înțelege diversitatea datelor, am utilizat `df.nunique()` pentru a verifica numărul de valori unice din fiecare coloană și `df[column].unique()` pentru a obține aceste valori. Am identificat hotelul cu cele mai puține și cele mai multe recenzii, folosind `df.ReviewsCount.min()` și `df.ReviewsCount.max()`, și am obținut detalii despre aceste hoteluri. De asemenea, am calculat numărul mediu și median al recenziilor folosind `df.ReviewsCount.mean()` și `df.ReviewsCount.median()`. Am aplicat o filtrare pentru a afișa hotelurile cu prețuri mai mici de 100 de euro pe noapte folosind `df[df.Price < 100]`, astfel obținând o listă a acestor hoteluri.

CODUL UTILIZAT:

```
#afisarea tuturor valorilor dintr-o coloana
print(df['Name'])
#SAU
print(df.Name)

#afisarea tipului de date dintr-o coloana
print(type(df['Name']))

#afisarea primelor 10 linii pentru coloana 'Name'
print(df.Name.head(10))

#afisarea ultimelor 10 linii pentru coloana 'Name'
print(df.Name.tail(10))

#afisarea valorilor pentru coloana 'Name'
print(df.Name.values)

#afisarea indexului pentru coloana 'Name'
print(df.Name.index)

#Statistici descriptive pentru coloane numerice
print(df.describe())
#Statistici descriptive pentru coloanele de tip string
print(df.describe(include = 'object'))

#afisarea datelor doar pt coloanele numerice
numeric_columns_df = df.select_dtypes(include = [np.number])
print(numeric_columns_df)
#afisarea denumirilor coloanelor numerice
print(numeric_columns_df.columns)
```

```
#Afisarea valorilor unice din fiecare coloana
print(df.unique())

#afisarea numarului de valori unice pt fiecare coloana
for column in df.columns:
    print(f"Coloana {column} are {df[column].nunique()} valori unice.")

#afisarea valorilor unice pt fiecare coloana
for column in df.columns:
    print(f"Valorile unice din coloana {column} sunt:"
          f" {df[column].unique()} ")

#afisarea statisticilor descriptive pentru coloana 'Price'
print(df.Price)
print(df.Price.describe())

#afisarea hotelurilor cu preturi sub 100 euro
print(df[df.Price < 100])

#afisarea unor statistici despre numarul de review-uri

#afisarea hotelului cu cel mai mic nr de review-uri
print("Hotelul care are cel mai mic numar de review-uri este", df.Name[df.ReviewsCount.min()], ", avand", df.ReviewsCount.min(), "review.")

#afisarea hotelului cu cel mai mare nr de review-uri
index_max_reviews = df.ReviewsCount.idxmax()
hotel_name = df.Name.loc[index_max_reviews]
max_reviews = df.ReviewsCount.loc[index_max_reviews]
print(f"Hotelul care are cel mai mare numar de review-uri este {hotel_name}, avand {max_reviews} review-uri.")
```

```
#afisarea numarului mediu de review-uri
print("Numarul mediu de review-uri:", df.ReviewsCount.mean())

#afisarea medianei numarului de review-uri.
print("Mediana numarului de review-uri:", df.ReviewsCount.median())
... .
```

REZULTATELE OBTINUTE:

```

0          BUNK Hotel Amsterdam
1          YOTEL Amsterdam
2          Multatuli Hotel
3          nhow Amsterdam Rai
4          Motel One Amsterdam
...
520        Stadslogement Bij de Sassenpoort
521        Mercure Hotel Zwolle
522        The Cabin at Zwolle Central
523        Hanze Hotel Zwolle
524        Campanile Hotel & Restaurant Zwolle
Name: Name, Length: 525, dtype: object
0          BUNK Hotel Amsterdam
1          YOTEL Amsterdam
2          Multatuli Hotel
3          nhow Amsterdam Rai
4          Motel One Amsterdam
...
520        Stadslogement Bij de Sassenpoort
521        Mercure Hotel Zwolle
522        The Cabin at Zwolle Central
523        Hanze Hotel Zwolle
524        Campanile Hotel & Restaurant Zwolle
Name: Name, Length: 525, dtype: object

```

```

<class 'pandas.core.series.Series'>
0          BUNK Hotel Amsterdam
1          YOTEL Amsterdam
2          Multatuli Hotel
3          nhow Amsterdam Rai
4          Motel One Amsterdam
...
5          INNSIDE by Meliá Amsterdam
6          Eden Hotel Amsterdam
7          citizenM Amsterdam South
8          The Alfred Hotel
9          Andaz Amsterdam Prinsengracht - a concept by Hyatt
Name: Name, dtype: object
515        Bed & Breakfast 'Aan de IJssel'
516        B&B de Luwe Cottage
517        Hotel Oldenburg
518        Hotel Fidder - Patrick's Whisky Bar
519        Buitengoed de Luwe
520        Stadslogement Bij de Sassenpoort
521        Mercure Hotel Zwolle
522        The Cabin at Zwolle Central
523        Hanze Hotel Zwolle
524        Campanile Hotel & Restaurant Zwolle
Name: Name, dtype: object
['BUNK Hotel Amsterdam' 'YOTEL Amsterdam' 'Multatuli Hotel'
 'nhow Amsterdam Rai' 'Motel One Amsterdam' 'INNSIDE by Meliá Amsterdam'
 'Eden Hotel Amsterdam' 'citizenM Amsterdam South' 'The Alfred Hotel'
 'Andaz Amsterdam Prinsengracht - a concept by Hyatt']

```

```

'Hyatt Regency Amsterdam' 'NH Collection Amsterdam Flower Market'
'ibis Styles Amsterdam Central Station' 'Leonardo Royal Hotel Amsterdam'
'Rho Hotel' 'Leonardo Boutique Museumhotels' 'Hotel V Nesplein'
'Kimpton De Witt Amsterdam, an IHG Hotel' 'Qbic Hotel WTC Amsterdam'
'Hotel Espresso' 'Hotel Central Park' 'The White Tulip Hostel'
'Swissotel Amsterdam'
'Holiday Inn Express Amsterdam - City Hall, an IHG Hotel'
'Hotel Atlantis Amsterdam' 'Holiday Home Green Resort Mooi Bemelen-19'
'Holiday Home Green Resort Mooi Bemelen-2'
'Holiday Home Green Resort Mooi Bemelen-12'
'Holiday Home Green Resort Mooi Bemelen-31'
'Holiday Home Green Resort Mooi Bemelen-15'
'Holiday Home Green Resort Mooi Bemelen-10'
'Holiday Home Green Resort Mooi Bemelen-13'
'Holiday Home Green Resort Mooi Bemelen-16' 'Resort Mooi Bemelen'
'Holiday Home Green Resort Mooi Bemelen-6'
'Holiday Home Green Resort Mooi Bemelen-3'
'Holiday Home Green Resort Mooi Bemelen-17'
'Holiday Home Green Resort Mooi Bemelen-21'
'Holiday Home Green Resort Mooi Bemelen-10'
'Holiday Home Green Resort Mooi Bemelen-4'
'Holiday Home Green Resort Mooi Bemelen-11'
'Holiday Home Green Resort Mooi Bemelen-32'
'Holiday Home Green Resort Mooi Bemelen-33'
'Holiday Home Green Resort Mooi Bemelen-20'

```

	ID	Price	ReviewsCount	Rating
count	525.000000	525.000000	525.000000	525.000000
mean	262.000000	141.992705	725.856356	8.328624
std	151.698715	73.385234	872.738966	0.705449
min	0.000000	43.870000	1.000000	3.700000
25%	131.000000	96.100000	229.000000	8.000000
50%	262.000000	124.700000	500.000000	8.400000
75%	393.000000	157.170000	846.000000	8.800000
max	524.000000	587.830000	7748.000000	10.000000
			Name ... State	
count			525 ... 525	
unique			522 ... 19	
top			Holiday Home Green Resort Mooi Bemelen-12 ... Limburg	
freq			2 ... 70	
			[4 rows x 4 columns]	
	ID	Price	ReviewsCount	Rating
0	0	86.76	778.0	8.4
1	1	167.94	500.0	8.1
2	2	143.69	1605.0	7.4
3	3	141.39	500.0	9.0
4	4	104.18	500.0	8.8
...
520	520	97.89	232.0	8.4
521	521	112.26	1402.0	7.7

```

522 522   67.35    149.0   7.2
523 523   85.68   1095.0   7.2
524 524   75.57   2071.0   6.4

[525 rows x 4 columns]
Index(['ID', 'Price', 'ReviewsCount', 'Rating'], dtype='object')
ID      522
Name     522
Type     190
Price    358
ReviewsCount 309
Rating   42
City     50
State    19
dtype: int64
Coloane ID are 525 valori unice.
Coloane Name are 522 valori unice.
Coloane Type are 190 valori unice.
Coloane Price are 358 valori unice.
Coloane ReviewsCount are 309 valori unice.
Coloane Rating are 42 valori unice.
Coloane City are 50 valori unice.
Coloane State are 19 valori unice.
Valoile unice din coloana ID sunt: [ 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17
 18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35
 36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53
 54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71
 72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89
 90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106
 107  108  109  110  111  112  113  114  115  116  117  118  119  120
 121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138
 139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156
 157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174
 175  176  177  178  179  180  181  182  183  184  185  186  187  188  189
 190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210
 211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228
 229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246
 247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263
 264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279
 280  281  282  283  284  285  286  287  288  289  290  291  292  293
 294  295  296  297  298  299  300  301  302  303  304  305
 306  307  308  309  310  311  312  313  314  315  316  317  318
 319  320  321  322  323  324  325  326  327  328  329  330  331
 332  333  334  335  336  337  338  339  340  341
 342  343  344  345  346  347  348  349  350  351  352  353  354
 355  356  357  358  359
 360  361  362  363  364  365  366  367  368  369  370  371  372
 373  374  375  376  377
 378  379  380  381  382  383  384  385  386  387  388  389  390
 391  392  393  394  395
 396  397  398  399  400  401  402  403  404  405  406  407  408
 409  410  411  412  413
 414  415  416  417  418  419  420  421  422  423  424  425  426
 427  428  429  430  431
 432  433  434  435  436  437  438  439  440  441  442  443  444
 445  446  447  448  449
 450  451  452  453  454  455  456  457  458  459  460  461  462
 463  464  465  466  467
 468  469  470  471  472  473  474  475  476  477  478  479  480
 481  482  483  484  485
 486  487  488  489  490  491  492  493  494  495  496  497  498
 499  500  501  502  503
 504  505  506  507  508  509  510  511  512  513  514  515  516
 517  518  519  520  521
 522  523  524]
Valoile unice din coloana Name sunt: ['BUNK Hotel Amsterdam' 'YOTEL Amsterdam' 'Multatuli Hotel'
 'nhow Amsterdam Rai' 'Motel One Amsterdam' 'INNSIDE by Meliá Amsterdam'
 'Eden Hotel Amsterdam' 'citizenM Amsterdam South' 'The Alfred Hotel'
 'Andaz Amsterdam Prinsengracht - a concept by Hyatt']

```

```

108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233
234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251
252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269
270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287
288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305
306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323
324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341
342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359
360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377
378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395
396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413
414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431
432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449
450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467
468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485
486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503
504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521
522 523 524]
Valoile unice din coloana Name sunt: ['BUNK Hotel Amsterdam' 'YOTEL Amsterdam' 'Multatuli Hotel'
 'nhow Amsterdam Rai' 'Motel One Amsterdam' 'INNSIDE by Meliá Amsterdam'
 'Eden Hotel Amsterdam' 'citizenM Amsterdam South' 'The Alfred Hotel'
 'Andaz Amsterdam Prinsengracht - a concept by Hyatt']

```

```

Valorile unice din coloana Type sunt: ['Bunk Pod for 2' 'Premium Double Room' 'Double Room'
 'Inhow Double or Twin Room with View' 'The Innside Guestroom'
 'Small Double Room' 'King Room' 'Standard Double Room' 'Observatory King'
 'Twin Room' 'Superior Double or Twin Room'
 'Queen Room - Disability Access' 'Standard Double or Twin Room'
 'Double or Twin Room' 'Comfort Double or Twin Room'
 '1 Queen or King Bed Essential Room' 'Standard Double Room - No Window'
 'Standard Twin Bunk with Shared Bathroom' 'Classic Double Room'
 'Holiday Home' 'Two-Bedroom House' 'Double Room with Shared Bathroom'
 'Two-Bedroom Apartment' 'One-Bedroom Apartment'
 'Small Double Room with Shared Bathroom'
 'Double Room with Private Bathroom' 'Superior, Guest room, 1 King'
 'Suite' 'Comfort Twin Room' 'Deluxe Double or Twin Room' 'Standard Suite'
 'Comfort Double Room with Shower' 'Standard Twin Room'
 'Superior Double Room with Sofa Bed' 'Superior Queen Room'
 'Twin Standard' 'Deluxe Double Room' 'Family Suite'
 'Small Double Room with Canal View' ' Deluxe double or Twin Room'
 'Basic Double Room' 'Standard Double Room with City View'

```

Valorile unice din coloana Price sunt: [86.76 167.94 143.69 141.39 104.18 155.35 220.66 156.27 139.21 495.42
345.51 323.3 195.78 124.7 206.56 168.93 241.09 380.79 88.91 142.91
74.73 134.71 233.18 190.39 101.52 230.8 291.87 311.63 130.22 78.5
129.33 214.82 76.34 76.43 91.16 113.74 175.12 92.19 103.36 104.93
111.54 125.73 90.06 120.7 67.42 79.93 97.89 43.87 145.94 69.16
96.17 107.77 126.85 210.85 215.53 225.89 83.78 92.15 96.99 103.28
110.01 117.88 264.51 77.25 84.94 92.37 60.67 135.82 159.5 98.34
106.69 131.14 123.75 134.43 126.49 132.24 155.19 99.14 98.47 112.62
160.75 120.43 71.84 121.25 70.95 72.35 144.6 56.19 230.35 62.84
67.4 68.57 73.61 100.86 116.01 141. 149.97 61.96 62.87 66.39
68.26 70.22 72.95 76.01 89.49 90.88 103.46 340.37 112.26 140.1
158.96 83.07 86.22 106.88 117.65 123.94 138.3 171.75 184.11 48.28

Valorile unice din coloana Rating sunt: [8.4 8.1 7.4 9. 8.8 8.5
7.3 8.9 8.5 8.6 7.9 8.
7.8 6.8 7. 10. 9.6 9.2
9.1 9.4 8.7 8.2 7.7 7.2
6.5 7.6 7.1 9.3 6.6 9.7
9.5 9.8 5.5 6.9 7.5 6.7
8.25263158 5. 6.2 6.4 3.7 8.175]

Valorile unice din coloana City sunt: ['Amsterdam Noord' 'Amsterdam City Center' 'Zuideramstel' 'Oud Zuid'
 'Oost' 'Oud West' 'Bemelen' 'Breda' 'Delft' 'Den Bosch' 'Strip'
 'Eindhoven City Centre' 'Stratum' 'Eindhoven' 'Woensel-Zuid' 'Tongelre'
 'Groningen' 'Haarlem' 'The Hague' 'The Hague City Centre' 'Laak'
 'Haage Hout' 'Segbroek' 'Hoofddorp' 'Leeuwarden' 'Maastricht' 'Wijck'
 'Jekerkwartier' 'Maastricht City Centre' 'Boschstraatkwartier'
 'Randwijck' 'Middelburg' 'Nijmegen' 'Roermond' 'Delfshaven' 'Centrum'
 'Feijenoord' 'Kralingen-Crooswijk' 'Scheveningen' 'City Centre' 'Utrecht'
 'Zuidwest' 'West' 'Zuid' 'Noordooost' 'Valkenburg' 'Vlissingen'
 'Voorthuizen' 'Zandvoort' 'Zwolle']

Valorile unice din coloana State sunt: [' Amsterdam' 'Limburg' 'North Brabant' 'South Holland' ' Eindhoven'
 'Eindhoven' 'Groningen' 'North Holland' 'The Hague' ' The Hague'
 'Friesland' 'Maastricht' ' Maastricht' 'Zeeland' ' Gelderland'
 ' Rotterdam' ' Utrecht' 'Utrecht' 'Overijssel']

0 86.76
1 167.94
2 143.69
3 141.39
4 104.18

522 67.35
523 85.68
524 75.57
Name: Price, Length: 525, dtype: float64
count 525.000000
mean 141.992705
std 73.385234
min 43.870000
25% 96.100000
50% 124.700000
75% 157.170000
max 587.830000
Name: Price, dtype: float64
ID Name ... City State
0 0 BUNK Hotel Amsterdam ... Amsterdam Noord Amsterdam
18 18 Qbic Hotel WTC Amsterdam ... Zuideramstel Amsterdam
20 20 Hotel Central Park ... Oud Zuid Amsterdam
50 50 Anja's House ... Breda North Brabant
S1 51 B&B de Druif ... Breda North Brabant
...
519 519 Buitenplaats de Luwte ... Zwolle Overijssel
520 520 Stadslogement Bij de Sassenpoort ... Zwolle Overijssel
522 522 The Cabin at Zwolle Centraal ... Zwolle Overijssel
523 523 Hanze Hotel Zwolle ... Zwolle Overijssel
524 524 Campanile Hotel & Restaurant Zwolle ... Zwolle Overijssel

```
[154 rows x 8 columns]
Hotelul care are cel mai mic numar de review-uri este YOTEL Amsterdam , avand 1.0 review.
Hotelul care are cel mai mare numar de review-uri este Grand Hotel Amrâth Kurhaus The Hague Scheveningen, avand 7748.0 review-uri.
Numarul mediu de review-uri: 725.8563559343021
Mediana numarului de review-uri: 500.0
```

❖ Interpretarea rezultatelor

Setul de date conține 525 de hoteluri, fiecare cu un ID unic, variind de la 0 la 524. Prețul mediu pe noapte este de 141.99 euro, cu o mare diversitate de opțiuni, de la 43.87 euro (preț minim) până la 587.83 euro (preț maxim). Jumătate dintre hoteluri au prețuri mai mici de 124.70 euro, iar 75% dintre ele au prețuri sub 157.17 euro, ceea ce sugerează o gamă variată de opțiuni de cazare. Numărul mediu de recenzii pe hotel este de 725.86, dar acest număr variază semnificativ, de la 1 recenzie pentru un hotel nou sau mai puțin cunoscut, până la 7748 recenzii pentru hoteluri foarte populare. De asemenea, majoritatea hotelurilor au între 500 și 1000 de recenzii, ceea ce indică o acoperire bună și un feedback relevant. Ratingul mediu este de 8.33, ceea ce sugerează o satisfacție generală destul de bună. Referitor la valorile unice, fiecare hotel are un ID unic, iar din cele 525 de înregistrări, 522 au nume unice, indicând că doar 3 hoteluri împart același nume. Există 190 de tipuri de camere de hotel diferite, ceea ce reflectă o diversitate de categorii. Prețurile sunt variate, cu 358 de valori unice, iar numărul de recenzii variază, având 309 valori unice. Ratingurile sunt mai restrânse, cu doar 42 de valori unice, iar orașele sunt diverse, cu 50 de locații diferite, în timp ce statele sunt doar 19, sugerând o concentrare geografică specifică.

9. Accesarea datelor cu LOC si ILOC

❖ Descrierea problemei

Se dorește utilizarea metodelor iloc și loc pentru accesarea și manipularea datelor din fisierul csv, astfel incat sa se extraga informatii relevante despre hotelurile listate pe Booking.com din Amsterdam, precum numele hotelurilor, datele unui anumit hotel, filtrarea acestora in functie de anumite coloane si selectarea inregistrarilor pe baza anumitor conditii.

❖ Informatii necesare pentru rezolvare

Pentru realizarea acestor operatii, avem nevoie de datele din fisierul CSV.

❖ Produs software/functii/metoda de calcul folosita

- **Produs software folosit:** PyCharm
- **Limbaj:** Python
- **Functii/Metode utilizate:**
 - *Iloc()* - permite selectarea rândurilor și coloanelor pe baza indexului numeric
 - *Loc()* - permite accesarea datelor folosind etichetele acestora

❖ Rezolvarea cu ajutorul produsului software

In cadrul acestui capitol, am realizat diferite operații de accesare și filtrare a datelor din setul de date stocat intr-un dataframe Pandas. Se afișează denumirile tuturor hotelurilor, datele pentru un hotel specific, numele și prețul pentru primele două înregistrări și primele 15

înregistrări. Sunt filtrate hotelurile cu rating > 8 și preț < 100 euro, iar apoi sunt prezentate înregistrările dintr-un interval de index specificat. Se afișează prețurile pentru toate înregistrările, ultima coloană și ultima înregistrare completă. În final, sunt afișate înregistrările unde statul este 'South Holland'.

CODUL UTILIZAT:

```
#afisarea denumirilor tuturor hotelurilor (valorile de pe coloana 'Name')
print("Denumirile tuturor hotelurilor din setul de date:")
print(df.iloc[:,1])

#afisarea datelor hotelului cu indexul 300
print("Hotelul cu indexul 300:")
print(df.iloc[300])

#afisarea numelui (1) si pretului (3) pentru primele 2 inregistrari
print("Afisarea numelui si pretului pt primele 2 inregistrari:")
print(df.iloc[[0,1], [1,3]])

#afisarea primelor 15 inregistrari
print("Afisarea primelor 15 inregistrari:")
print(df.iloc[0:16])

#afisarea hotelurilor cu rating mai mare decat 8 si pretul sub 100 euro
print("Afisarea hotelurilor cu rating mai mare decat 8 si pretul sub 100 euro: ")
print(df.loc[(df.Rating > 8) & (df.Price < 100)])

#afisarea inregistrarilor de la 10 la 20 dupa eticheta, si apoi dupa index
print("Afisarea inregistrarilor de la 10 la 20 dupa eticheta: ")
print(df.loc[10:20])
print("Afisarea inregistrarilor de la 10 la 20 dupa index: ")
print(df.iloc[10:21])
```

```
#Afisarea preturilor pentru fiecare inregistrare
print("Afisarea preturilor pentru fiecare inregistrare:")
print(df.loc[:, 'Price'])

#Afisarea ultimei coloane (State) pentru fiecare inregistrare
print("Afisarea ultimei coloane:")
print(df.iloc[:, -1])

#Afisarea ultimei inregistrari
print("Afisarea ultimei inregistrari:")
print(df.iloc[-1, :])

#Afisarea inregistrarilor unde statul este 'South Holland'
print("Afisarea hotelurilor din statul South Holland:")
print(df.loc[df.State == 'South Holland'])
```

REZULTATELE OBTINUTE:

Denumirile tuturor hotelurilor din setul de date:		
0	BUNK Hotel Amsterdam	
1	YOTEL Amsterdam	
2	Multatuli Hotel	
3	nhow Amsterdam Rai	
4	Motel One Amsterdam	
	...	
520	Stadslogement Bij de Sassenpoort	
521	Mercure Hotel Zwolle	
522	The Cabin at Zwolle Centraal	
523	Hanze Hotel Zwolle	
524	Campanile Hotel & Restaurant Zwolle	
Name:	Name, Length: 525, dtype: object	
Hotelul cu indexul 300:		
ID	300	
Name	Van der Valk Hotel Nijmegen-Lent	
Type	Standard Double Room	
Price	110.46	
ReviewsCount	500.0	
Rating	8.8	
City	Nijmegen	
State	Gelderland	
Name:	300, dtype: object	

Afisarea numelui si pretului pt primele 2 inregistrari:		
0	Name	Price
0	BUNK Hotel Amsterdam	86.76
1	YOTEL Amsterdam	167.94
Afisarea primelor 15 inregistrari:		
ID	...	State
0	0	Amsterdam
1	1	Amsterdam
2	2	Amsterdam
3	3	Amsterdam
4	4	Amsterdam
5	5	Amsterdam
6	6	Amsterdam
7	7	Amsterdam
8	8	Amsterdam
9	9	Amsterdam
10	10	Amsterdam
11	11	Amsterdam
12	12	Amsterdam
13	13	Amsterdam
14	14	Amsterdam
15	15	Amsterdam

[16 rows x 8 columns]	[11 rows x 8 columns]
Afisarea hotelurilor cu rating mai mare decat 8 si pretul sub 100 euro:	Afisarea inregistrarilor de la 10 la 20 dupa index:
ID ... Name ... City ... State	ID ... State
0 0 BUNK Hotel Amsterdam ... Amsterdam Noord ... Amsterdam	10 10 ... Amsterdam
50 50 Anja's House ... Breda North Brabant	11 11 ... Amsterdam
51 51 B&B de Druif ... Breda North Brabant	12 12 ... Amsterdam
54 54 Boutique Hotel Het Scheepshuys ... Breda North Brabant	13 13 ... Amsterdam
55 55 B&B de Bievangh ... Breda North Brabant	14 14 ... Amsterdam
...	15 15 ... Amsterdam
516 516 B&B de Luwte Cottage ... Zwolle Overijssel	16 16 ... Amsterdam
517 517 Hotel Oldenburg ... Zwolle Overijssel	17 17 ... Amsterdam
518 518 Hotel Fidder - Patrick's Whisky Bar ... Zwolle Overijssel	18 18 ... Amsterdam
519 519 Buitenplaats de Luwte ... Zwolle Overijssel	19 19 ... Amsterdam
520 520 Stadslogement Bij de Sassenpoort ... Zwolle Overijssel	20 20 ... Amsterdam
[93 rows x 8 columns]	[11 rows x 8 columns]
Afisarea inregistrarilor de la 10 la 20 dupa etichete:	Afisarea preturilor pentru fiecare inregistrare:
ID ... State	0 86.76
10 10 ... Amsterdam	1 167.94
11 11 ... Amsterdam	2 143.69
12 12 ... Amsterdam	3 141.39
13 13 ... Amsterdam	4 104.18
14 14 ... Amsterdam	...
15 15 ... Amsterdam	520 97.89
16 16 ... Amsterdam	521 112.26
17 17 ... Amsterdam	522 67.35
18 18 ... Amsterdam	523 85.68
19 19 ... Amsterdam	524 75.57
20 20 ... Amsterdam	Name: Price, Length: 525, dtype: float64
Afisarea ultimei coloane:	Afisarea hotelurilor din statul South Holland:
0 Amsterdam	ID ... Name ... City ... State
1 Amsterdam	70 70 Hotel Arsenala Delft ... Delft South Holland
2 Amsterdam	71 71 The Student Hotel Delft ... Delft South Holland
3 Amsterdam	72 72 WestCord Hotel Delft ... Delft South Holland
4 Amsterdam	73 73 Hotel de Koophandel ... Delft South Holland
...	74 74 Casa Julia ... Delft South Holland
520 Overijssel	75 75 Luxurious loft with view walk city center ... Delft South Holland
521 Overijssel	76 76 De Vliegende Vos het geboortehuis van Johannes... ... Delft South Holland
522 Overijssel	77 77 City Center Penthouse With Balcony Delft ... Delft South Holland
523 Overijssel	78 78 Hotel Grand Canal ... Delft South Holland
524 Overijssel	79 79 Hampshire Hotel - Delft Centre ... Delft South Holland
Name: State, Length: 525, dtype: object	80 80 Hotel Royal Bridges ... Delft South Holland
Afisarea ultimei inregistrari:	81 81 Hotel de Plataan Delft Centrum ... Delft South Holland
ID 524	82 82 Buitengoed De Uylenburg ... Delft South Holland
Name Campanile Hotel & Restaurant Zwolle	83 83 ibis Styles Delft City Centre ... Delft South Holland
Type Double Room	84 84 Hotel Johannes Vermeer Delft ... Delft South Holland
Price 75.57	85 85 Lux Loft Apt View Walk city center ... Delft South Holland
ReviewsCount 2071.0	86 86 Shanghai Hotel Holland ... Delft South Holland
Rating 6.4	87 87 Best Western Museumhotels Delft ... Delft South Holland
City Zwolle	88 88 Hotel Bridges House Delft ... Delft South Holland
State Overijssel	89 89 Campanile Hotel & Restaurant Delft ... Delft South Holland
Name: 524, dtype: object	[20 rows x 8 columns]

❖ Interpretarea rezultatelor

Observăm că instrucțiunile loc și iloc sunt foarte utile în extragerea informațiilor din setul de date, facilitând observarea anumitor elemente. De exemplu, am afișat toate denumirile hotelurilor, precum și datele pentru un hotel specific luat după index. Am afișat un anumit număr de înregistrări, am filtrat înregistrările pe baza unor condiții specifice și am vizualizat datele acestora. De asemenea, am obținut hoteluri dintr-un anumit oraș sau regiune, facilitând astfel analiza și observarea datelor particulare, într-un mod foarte structurat și eficient.

10. Gruparea și agregarea datelor

❖ Descrierea problemei

Scopul acestui capitol este de a analiza și agrega datele referitoare la hoteluri, având ca obiectiv obținerea unor statistici relevante despre tipurile de camere, prețuri, recenzii și ratinguri. Datele trebuie grupate în funcție de diferite caracteristici, cum ar fi tipul camerei (de exemplu, apartament, cameră standard etc.), orașul sau statul, iar apoi se vor calcula statistici descriptive

pentru fiecare grup. De asemenea, este important să se identifice și să se salveze aceste rezultate pentru utilizări ulterioare.

❖ ***Informatii necesare pentru rezolvare***

Pentru rezolvarea problemei, este necesar setul de date cu informații despre hoteluri, care trebuie să includă cel puțin următoarele coloane: Type, Price, ReviewsCount, Rating, State, City.

Scopul este să se grupeze datele în funcție de coloanele relevante și să se calculeze statistici descriere (media, suma, mediana, etc.) pentru fiecare grup.

Se dorește obținerea următoarelor rezultate:

- Identificarea numărului de apartamente din setul de date;
- Afisarea statisticilor de bază pentru fiecare tip de cameră (prețul mediu, suma recenziilor, valoarea mediană a rating-urilor);
- Determinarea numărului de tipuri de camere disponibile pentru fiecare oraș sau stat;
- Calcularea prețului mediu și a altor statistici pentru fiecare grup;

❖ ***Produs software/functii/metoda de calcul folosită***

- **Produs software folosit:** PyCharm
- **Limbaj:** Python
- **Functii/Metode utilizate:**
 - *groupby()*: Utilizat pentru a grupa datele în funcție de una sau mai multe coloane (de exemplu, Type, State, City).
 - *first()*: Permite obținerea primei înregistrări pentru fiecare grup.
 - *agg()*: Aplică funcții agregate pentru mai multe coloane simultan, cum ar fi calcularea mediei, sumei și medianei.
 - *count()*: Numără elementele din fiecare grup.
 - *mean()*: Calculează media valorilor pentru un grup.
 - *sum()*: Calculează suma valorilor pentru un grup.
 - *max()* și *min()*: Permite obținerea valorii maxime și minime pentru un grup.
 - *to_csv()*: Salvează rezultatele într-un fișier CSV pentru utilizare ulterioară.

❖ ***Rezolvarea cu ajutorului produsului software***

Rezolvarea problemei s-a realizat utilizând Python și biblioteca Pandas, prin aplicarea funcțiilor de grupare și agregare pe setul de date. Inițial, datele au fost grupate după tipul camerei utilizând funcția *groupby()*, iar pentru a obține statistică relevante pentru fiecare grup, s-au folosit funcții precum *first()* pentru a extrage prima înregistrare din fiecare grup, *count()* pentru a număra camerele disponibile, și *agg()* pentru a calcula multiple statistică descriptive (media prețului, suma recenziilor, mediana rating-urilor). De asemenea, pentru a obține valori mai detaliate, au fost utilizate funcțiile *mean()*, *max()*, *min()*, iar rezultatele agregate au fost salvate într-un fișier CSV cu ajutorul metodei *to_csv()*. Astfel, prin combinarea acestor funcții, am reușit să obținem o analiză detaliată a datelor, care a inclus informații despre prețuri, recenzi și ratinguri pentru fiecare tip de cameră, oraș și stat.

CODUL UTILIZAT:

```
#Grupari
#afisarea cheilor de grupare pentru 'Type' (tipul de camera)
print("Cheile de grupare sunt:")
print(df.groupby(['Type']).groups.keys())

#Sa se numere camerele de tip apartament
nr_apartamente = len(df.groupby(['Type']).groups['Apartment'])
print("Numarul de apartamente este %d" % nr_apartamente)

#afisarea primei inregistrari pentru fiecare valoare din 'Type' (fiecare tip de camera)
print("Prima inregistrare aferenta fiecarei valoare din 'Type':")
print(df.groupby('Type').first())

#afisarea pretului mediu pentru fiecare stat
print("Pretul mediu per stat:")
print(df.groupby(by = 'State')[['Price']].mean())

#afisarea pretului maxim in functie de tipul camerei
print("Pretul maxim in functie de tipul camerei:")
print(df.groupby(by = 'Type')[['Price']].max())

#afisarea celui mai mic rating pentru fiecare oras
print("Cel mai mic rating pentru fiecare oras:")
print(df.groupby(by = 'City')[['Rating']].min())

```

```

#afisarea numerului de tipuri de camere pentru fiecare regiune si oras
print('Numarul de tipuri de camere de hotel pentru fiecare regiune si oraș:')
print(df.groupby(['State', 'City'])['Type'].count())

#Sa se grupeze dupa tipul de camera si sa se calculeze statistici pentru fiecare grup (pretul mediu,
# suma nr de review-uri si valoarea mediana a ratingurilor)
print("Calcularea statisticilor pentru fiecare tip de camera:")
print(df.groupby(['Type']).agg({'Price': 'mean',
                               'ReviewsCount': 'sum',
                               'Rating': 'median'}))

#Salvarea agregarii intr-un fisier csv
df_aggregate = df.groupby(['Type']).agg({'Price': 'mean',
                                         'ReviewsCount': 'sum',
                                         'Rating': 'median'})
df_aggregate.to_csv('agregate_project.csv')

#Statistici descriptive multiple pt fiecare grup
print("Statistici multiple pentru fiecare grup:")
print(df.groupby(['Type']).agg({'Price': ['min', 'max','mean'],
                               'ReviewsCount': 'sum',
                               'Rating': ['median', 'min', 'max']}))

```

REZULTATELE OBTINUTE:

```
Cheile de grupare sunt:
dict_keys([' Deluxe double or Twin Room', '1 Queen or King Bed Essential Room', '2-person Premium Hotelroom',
Numarul de apartamente este 2
Prima inregistrare aferenta fiecarei valoare din 'Type':

```

Type	ID	...	State
Deluxe double or Twin Room	80	...	South Holland
1 Queen or King Bed Essential Room	17	...	Amsterdam
2-person Premium Hotelroom	498	...	North Holland
6 Person Room with Private Bathroom and Shower	256	...	Maastricht
Apartment	356	...	Rotterdam
...
Two-Bedroom Chalet	438	...	Zeeland
Two-Bedroom House	33	...	Limburg
Two-Bedroom Suite	207	...	The Hague
bunk	405	...	Utrecht
nhow Double or Twin Room with View	3	...	Amsterdam

Pretul mediu per stat:			Pretul maxim in functie de tipul camerei:		
State			Type		
Amsterdam	190.474000		Deluxe double or Twin Room	96.99	
Eindhoven	92.409167		1 Queen or King Bed Essential Room	380.79	
Maastricht	325.690000		2-person Premium Hotelroom	133.82	
Rotterdam	127.015600		6 Person Room with Private Bathroom and Shower	161.66	
The Hague	138.649524		Apartment	302.29	
Utrecht	133.144348		
Eindhoven	68.260000		Two-Bedroom Chalet	208.56	
Friesland	102.597500		Two-Bedroom House	340.37	
Gelderland	145.259333		Two-Bedroom Suite	206.92	
Groningen	119.162800		bunk	112.26	
Limburg	169.975857		nhow Double or Twin Room with View	141.39	
Maastricht	200.760000		Name: Price, Length: 190, dtype: float64		
North Brabant	114.542093				
North Holland	133.091692				
Overijssel	106.530952				
South Holland	124.487000				
The Hague	138.443846				

Cel mai mic rating pentru fiecare oras:	
City	Rating
Amsterdam City Center	7.0
Amsterdam Noord	8.1
Bemelen	8.0
Boschstraatkwartier	7.5
Breda	6.5
Centrum	8.0
City Centre	8.2
Delfshaven	9.1
Delft	7.1
Den Bosch	6.5
Eindhoven	8.1
Eindhoven City Centre	8.0
Feijenoord	8.3
Groningen	8.0
Haagse Hout	8.5
Haarlem	5.5
Hoofddorp	6.9

Numărul de tipuri de camere de hotel pentru fiecare regiune și oraș:		
State	City	Count
Amsterdam	Amsterdam City Center	12
	Amsterdam Noord	2
	Oost	1
	Oud West	1
	Oud Zuid	4
	Zuideramstel	5
Eindhoven	Eindhoven City Centre	16
	Stratum	1
	Strijp	5
	Tongelre	1
	Woensel-Zuid	1
Maastricht	Boschstraatkwartier	2
	Jekerkwartier	2
	Maastricht City Centre	3
	Randwijk	1
	Wijck	7
Rotterdam	Centrum	14
	Delfshaven	1
	Feijenoord	8

Calcularea statisticilor pentru fiecare tip de camera:				
Type	Price	...	Rating	...
Deluxe double or Twin Room	96.990000	...	8.30	
1 Queen or King Bed Essential Room	380.790000	...	8.80	
2-person Premium Hotelroom	133.820000	...	7.80	
6 Person Room with Private Bathroom and Shower	161.660000	...	8.30	
Apartment	224.340000	...	8.95	
...
Two-Bedroom Chalet	151.136667	...	8.00	
Two-Bedroom House	193.951429	...	9.20	
Two-Bedroom Suite	206.920000	...	8.50	
bunk	112.260000	...	8.20	
nhow Double or Twin Room with View	141.390000	...	9.00	

Statistică multiple pentru fiecare grup:						
Type	Price	...	Rating	min	max	...
Deluxe double or Twin Room	96.99	96.99	...	8.3	8.3	...
1 Queen or King Bed Essential Room	380.79	380.79	...	8.8	8.8	
2-person Premium Hotelroom	133.82	133.82	...	7.8	7.8	
6 Person Room with Private Bathroom and Shower	161.66	161.66	...	8.3	8.3	
Apartment	146.39	302.29	...	8.6	9.3	
...
Two-Bedroom Chalet	81.91	208.56	...	7.2	8.7	
Two-Bedroom House	79.37	340.37	...	8.1	9.8	
Two-Bedroom Suite	206.92	206.92	...	8.5	8.5	
bunk	112.26	112.26	...	8.2	8.2	
nhow Double or Twin Room with View	141.39	141.39	...	9.0	9.0	

[190 rows x 7 columns]

Process finished with exit code 0

❖ Interpretarea rezultatelor

In urma gruparii, se poate observa o varietate de tipuri de camere de hotel, de la camere duble deluxe sau twin, camere esențiale cu un pat queen sau king, apartamente, camere pentru 6 persoane cu baie privată, până la camere de tip „bunk” și „nhow Double or Twin Room with View”.

Ulterior, am realizat o analiză pe regiuni și orașe, evidențiind numărul de camere disponibile în diverse locații, de exemplu, Amsterdam, Eindhoven, Maastricht, Rotterdam, The Hague și Utrecht. Aceste informații sunt utile pentru a înțelege distribuția tipurilor de camere pe diferite zone. Analizând numarul de tipuri de camere per regiune și oraș: în Amsterdam sunt disponibile camere în diferite locații (Amsterdam City Center, Amsterdam Noord, etc.), iar în Rotterdam sunt și diverse opțiuni de camere distribuite în zone precum Centrum și Delfshaven.

Prețurile variază în funcție de tipul camerei, iar evaluările oferite de clienți sunt un alt aspect important. De exemplu, o cameră „Deluxe Double or Twin Room” are un preț mediu de 96.99 EUR și o evaluare de 8.3, în timp ce un „Apartment” poate ajunge la 302.29 EUR cu o evaluare de 8.95.

Prețurile medii pentru camere diferă în funcție de locație. De exemplu, în Amsterdam prețul mediu este de 190.47 EUR, în Rotterdam este de 127.02 EUR, iar în Maastricht este de 325.69 EUR.

Am identificat o varietate de tipuri de camere de hotel, de la camere duble deluxe sau twin, camere esențiale cu un pat queen sau king, apartamente, camere pentru 6 persoane cu baie privată, până la camere de tip „bunk” și „nhow Double or Twin Room with View”, evidențiind distribuția acestora în diferite regiuni și orașe precum Amsterdam, Eindhoven, Maastricht, Rotterdam, The Hague și Utrecht. În Amsterdam, camerele sunt disponibile în diverse locații, cum ar fi Amsterdam City Center și Amsterdam Noord, iar în Rotterdam, opțiunile sunt distribuite în zone precum Centrum și Delfshaven. Prețurile variază în funcție de tipul camerei, iar evaluările clienților oferă un indicator important al satisfacției: de exemplu, o „Deluxe Double or Twin Room” are un preț mediu de 96.99 EUR și un rating de 8.3, în timp ce un „Apartment” poate ajunge la 302.29 EUR cu un rating de 8.95. De asemenea, ai analizat diferențele de preț între orașe, unde Amsterdam are un preț mediu de 190.47 EUR, Rotterdam 127.02 EUR și Maastricht, unul dintre cele mai scumpe orașe, 325.69 EUR. Pe măsură ce prețul crește, numărul de recenzii este mai mare, indicând popularitatea sporită a anumitor tipuri de camere, iar ratingurile generale sugerează că turiștii sunt dispuși să plătească mai mult pentru confort sporit. În plus, ratingurile minime variază semnificativ între locații, cu zone precum Delfshaven și Laak având scoruri ridicate (9.1), în timp ce Vlissingen (3.7), Middelburg (5.0) și Haarlem (5.5) indică o satisfacție mai scăzută, reflectând diferențele de ofertă și cerere. Aceste concluzii subliniază tendințele pieței hoteliere, unde orașele mari au prețuri mai ridicate datorită cererii mari, în timp ce zonele mai puțin populare tind să aibă scoruri mai mici și o ofertă mai slabă de hoteluri.

11. Reprezentări grafice

❖ Descrierea problemei

În analiza pieței hoteliere, este esențial să vizualizăm și să înțelegem distribuția rating-urilor oferite de clienți, distribuția geografică a hotelurilor și rating-urile medii pe orașe. Prin utilizarea tehniciilor de vizualizare grafică, putem extrage informații relevante despre calitatea serviciilor hoteliere, preferințele consumatorilor și tendințele pieței în diferite regiuni.

❖ *Informatii necesare pentru rezolvare*

Pentru a realiza aceste vizualizări, este necesar un set de date care să conțină cel puțin următoarele coloane: Rating, State, City, Type.

❖ *Produs software/functii/metoda de calcul folosita*

- **Produs software folosit:** PyCharm
- **Limbaj:** Python
- **Functii/Metode utilizate:**
 - *plt.hist()* – pentru crearea histogramelor.
 - *plt.pie()* – pentru diagramele circulare.
 - *df.groupby().mean()* și *df.groupby().first()* – pentru calcularea statisticilor necesare.
 - *plt.bar()* – pentru reprezentarea rating-ului mediu pe orașe.

❖ *Rezolvarea cu ajutorul produsului software*

Pentru rezolvarea acestei analize s-a utilizat limbajul Python împreună cu biblioteca Matplotlib, permitând reprezentarea grafică a datelor într-un mod intuitiv și eficient. În primul rând, pentru a analiza distribuția rating-urilor, s-a folosit o histogramă generată cu *plt.hist()*, evidențiind frecvența fiecărui interval de scoruri acordate hotelurilor, unde s-a observat o concentrare majoritară a rating-urilor între valorile 7 și 9. În continuare, distribuția geografică a hotelurilor a fost reprezentată printr-o diagramă circulară utilizând *plt.pie()*, care a permis identificarea regiunilor cu cele mai multe hoteluri, printre care se remarcă Limburg și North Holland. Pentru a analiza diferențele de percepție a serviciilor hoteliere la nivel urban, s-a calculat media rating-ului pe orașe utilizând *df.groupby('City')['Rating'].mean()*, iar rezultatele au fost afișate printr-un grafic de tip bară cu *plt.bar()*, evidențiind variațiile calității percepute în funcție de locație. Aceste reprezentări grafice oferă o perspectivă clară asupra tendințelor pieței hoteliere, fiind utile în luarea deciziilor strategice pentru investitori și manageri din industrie.

CODUL UTILIZAT:

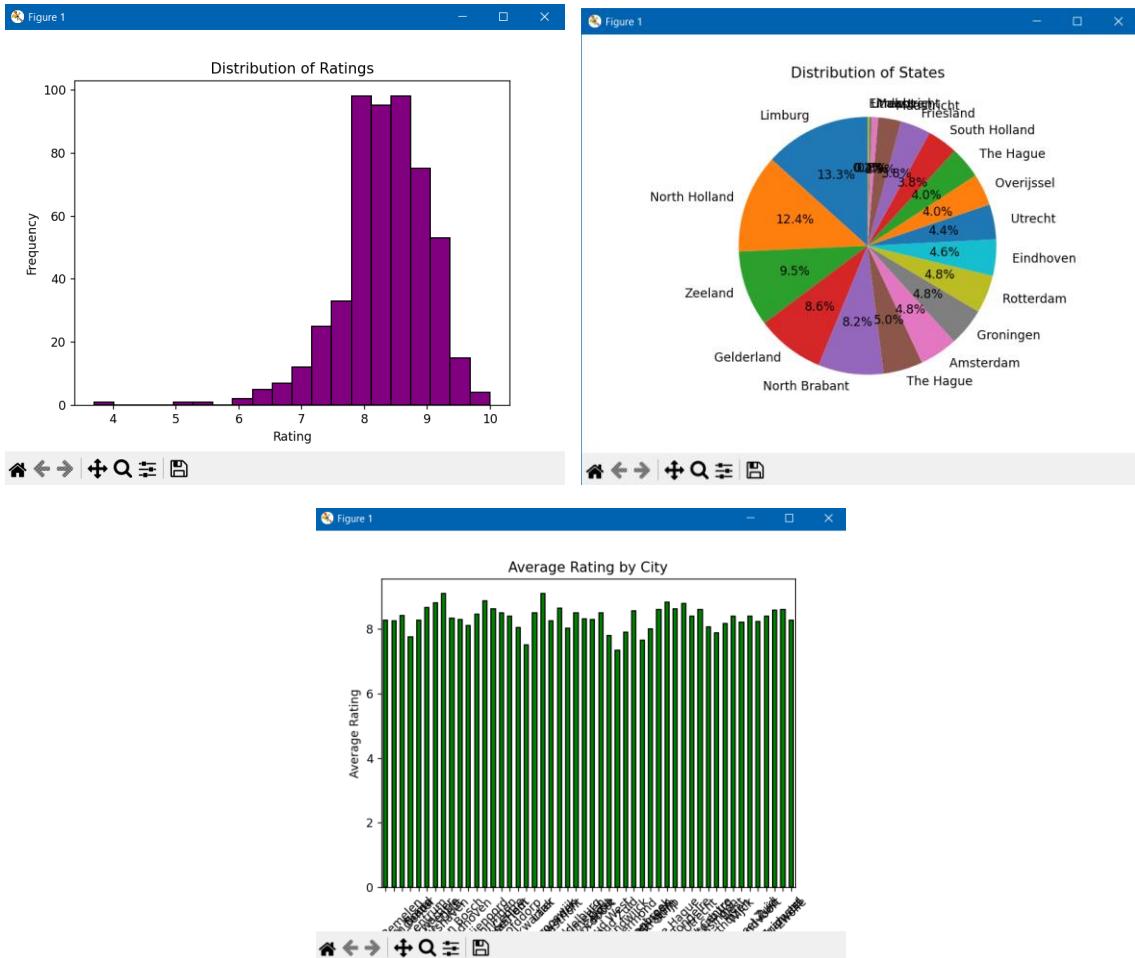
```
#Grafice
#Histograma/Distributia ratingurilor

plt.hist(df['Rating'], bins=20, color='purple', edgecolor='black')
plt.title('Distribution of Ratings')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.show()

#Distributia statelor
state_counts = df['State'].value_counts()
plt.pie(state_counts, labels=state_counts.index, autopct='%1.1f%%', startangle=90)
plt.title("Distribution of States")
plt.show()

#Ratingul mediu in functie de oras
average_rating_by_city = df.groupby('City')['Rating'].mean()
average_rating_by_city.plot(kind='bar', color='green', edgecolor='black')
plt.title('Average Rating by City')
plt.xlabel('City')
plt.ylabel('Average Rating')
plt.xticks(rotation=45)
plt.show()
```

REZULTATELE OBTINUTE:



❖ Interpretarea rezultatelor

Histograma prezintă distribuția ratingurilor hotelurilor, evidențiind o asimetrie spre stânga, ceea ce indică faptul că majoritatea ratingurilor sunt ridicate, situându-se între valorile 7 și 9. Cele mai frecvente evaluări se regăsesc în intervalul 8-9, ceea ce reflectă o tendință generală a hotelurilor de a primi recenzii pozitive. Ratingurile sub 6 sunt foarte rare, ceea ce sugerează că hotelurile analizate sunt, în general, bine cotate. De asemenea, se observă o ușoară scădere a frecvenței după valoarea 9, indicând că puține hoteluri reușesc să obțină un scor apropiat de 10. Această analiză confirmă că majoritatea hotelurilor oferă servicii de calitate și sunt apreciate pozitiv de clienți, în timp ce ratingurile slabe sunt excepționale.

Diagrama cu bare ilustrează ratingul mediu al hotelurilor pentru fiecare oraș analizat și oferă câteva concluzii importante. În general, ratingurile sunt ridicate, majoritatea orașelor având un rating mediu de peste 8, ceea ce indică faptul că hotelurile oferă servicii de calitate și sunt apreciate de clienți. Totuși, există variații între orașe, unele având ratinguri medii de peste 8.5, iar altele având ratinguri mai apropiate de 8. De asemenea, nu se observă ratinguri foarte scăzute (sub 6), ceea ce sugerează o calitate bună a hotelurilor în toate regiunile analizate. Cu toate acestea, etichetele de pe axa X sunt greu de citit din cauza unghiului acestora, dar în general, se

poate observa că ratingurile sunt relativ apropriate între orașe. În ansamblu, analiza sugerează că hotelurile din toate orașele studiate sunt bine cotate, cu diferențe minore între locații.

Diagrama tip pie chart ilustrează distribuția hotelurilor pe regiuni (state) dintr-o anumită zonă, evidențiind câteva observații importante. Limburg detine cea mai mare pondere, cu 13.3%, ceea ce sugerează că această regiune găzduiește cele mai multe hoteluri comparativ cu celelalte. North Holland și Zeeland urmează cu 12.4% și respectiv 9.5%, indicând o prezență semnificativă a hotelurilor în aceste zone. Celelalte regiuni au procente relativ echilibrate, cu valori între 4% și 9%, ceea ce sugerează o distribuție uniformă a hotelurilor pe întregul teritoriu analizat. Regiunile cu cele mai mici ponderi sunt greu de citit din cauza suprapunerii textului, dar par să aibă sub 2%, ceea ce indică o prezență mai redusă a hotelurilor în acele zone. În general, distribuția hotelurilor nu este uniformă, iar unele regiuni, precum Limburg și North Holland, au o densitate mai mare de hoteluri, ceea ce ar putea reflecta o cerere mai mare în aceste locații.

12. Prelucrarea seturilor de date utilizând MERGE

❖ *Descrierea problemei*

Scopul acestei funcții este de a îmbina două seturi de date referitoare la hoteluri pentru a obține o imagine completă asupra acestora. Primul set conține informații despre hoteluri, cum ar fi ID-ul, numele, tipul camerei, prețul, orașul și regiunea în care se află, iar al doilea set conține date despre numărul de recenzii și ratingul fiecărui hotel. Prin utilizarea operațiunii de merge (sau join) pe baza coloanei comune ID, se dorește obținerea unui set de date integrat care să includă atât informațiile despre hoteluri, cât și detaliile referitoare la recenzii și evaluări.

❖ *Informatii necesare pentru rezolvare*

Pentru a rezolva această problemă, sunt necesare două seturi de date:

- Un set de date care să includă coloanele ID, Name, Type, Price, City, State.
- Un alt set de date care să includă coloanele ID, ReviewsCount, Rating.

❖ *Produs software/functii/metoda de calcul folosită*

- **Produs software folosit:** PyCharm
- **Limbaj:** Python
- **Functii/Metode utilizate:**

Rezolvarea acestei probleme a fost realizată folosind limbajul Python și biblioteca pandas, care permite manipularea eficientă a datelor. Funcțiile utilizate includ:

- `pd.read_csv()` – pentru încărcarea datelor din fișiere CSV.
- `merge()` – pentru îmbinarea celor două seturi de date pe baza coloanei ID. Metoda `how='inner'` a fost utilizată pentru a păstra doar valorile comune între cele două tabele, astfel încât să fie păstrate doar acele hoteluri care apar în ambele seturi de date.

❖ *Rezolvarea cu ajutorul produsului software*

Seturile de date au fost încărcate utilizând funcția `pd.read_csv()`, specificând doar coloanele relevante pentru fiecare set. După aceasta, am folosit funcția `merge()` pentru a îmbina cele două tabele pe baza coloanei ID, astfel încât să obținem un singur DataFrame, `df_1`, care

conține atât informațiile despre hoteluri, cât și datele referitoare la recenzii și ratinguri. Astfel, am reușit să centralizăm toate informațiile necesare pentru analiza pieței hoteliere într-un singur tabel, gata pentru prelucrare și vizualizare ulterioară.

CODUL UTILIZAT:

```
#Prelucrarea setului de date cu MERGE

df_date = pd.read_csv('filepath_or_buffer: 'HotelDataset_Final.csv', usecols = ['ID', 'Name', 'Type', 'Price', 'City', 'State'])
df_reviewuri = pd.read_csv('filepath_or_buffer: 'HotelDataset_Final.csv', usecols = ['ID', 'ReviewsCount', 'Rating'])

df_1 = df_date.merge(df_reviewuri, on='ID', how='inner')
print(df_1.columns)
```

REZULTATELE OBTINUTE:

```
Index(['ID', 'Name', 'Type', 'Price', 'City', 'State', 'ReviewsCount',
       'Rating'],
      dtype='object')

Process finished with exit code 0
```

❖ Interpretarea rezultatelor

Setul de date combinat conține 190 de rânduri și 7 coloane, fiecare rând reprezentând un hotel. Coloanele includ informații precum ID-ul hotelului, numele, tipul camerei, prețul, locația (oraș și stat), numărul de recenzii și ratingul acordat de clienți. Aceste date permit efectuarea unor analize detaliate despre performanța hotelurilor, compararea prețurilor și a ratingurilor între diferite orașe sau regiuni, și evaluarea tendințelor în industria hotelieră.

PARTEA 2 – PROGRAMARE SAS

1. Crearea unui set de date SAS din fisiere externe

❖ Descrierea problemei

Pentru a analiza performanța hotelurilor listate pe Booking.com, am importat și am creat un set de date dintr-un fișier extern CSV care conține informații despre hoteluri, precum ID-ul, numele, tipul, prețul, numărul de recenzii, ratingul și locația acestora. Prin definirea variabilelor și încărcarea datelor în SAS, am pregătit setul pentru analiza ulterioară.

❖ Informatii necesare pentru rezolvare

Pentru analiză sunt necesare date detaliate despre fiecare unitate de cazare, precum numele hotelului, tipul camerei, prețul căzării (în EUR), numărul total de recenzii, ratingul mediu acordat de utilizatori, precum și locația geografică (oraș și stat). Aceste informații sunt esențiale pentru evaluarea factorilor care influențează performanța hotelurilor și pentru identificarea oportunităților pe piața olandeză.

❖ *Produs software/functie/metode de calcul folosite*

- **Produs software folosit:** SAS Studio
- **Limbaj:** SAS
- **Functii/Metode utilizate:**
 - data - creează un nou set de date SAS (în cazul de față, hoteluri).
 - infile - specifică fișierul extern CSV pentru importul datelor.
 - dlm – defineste delimitatorul pentru campurile din setul de date.
 - dsd - activează modul de citire „delimiter-separated data”, tratând corect ghilimelele și valorile lipsă.
 - firstobs=2 – indică faptul că importul începe de la a doua linie.
 - length - definește tipul și lungimea variabilelor.
 - input - specifică ordinea și tipul variabilelor ce se citesc din fișier.
 - proc contents - generează un raport cu structura și descrierea setului de date SAS.
 - title - adaugă un titlu pentru raportul generat.

❖ *Rezolvarea cu ajutorului produsului software*

Am creat un set de date SAS prin importul unui fișier CSV ce conține informații despre hoteluri, folosind delimitatorul virgulă și omisiunea primei linii de antet. Pentru fiecare coloană s-au definit tipurile și lungimile variabilelor (numeric și caractere), asigurând o încărcare corectă a datelor în tabelul hoteluri. Ulterior, s-a utilizat procedura proc contents pentru a vizualiza structura setului de date și detaliile variabilelor, confirmând importul corect și oferind o descriere a datelor disponibile pentru analiză.

CODUL UTILIZAT:

```
*Crearea setului de date SAS;
data hoteluri;
infile '/home/u64207962/HotelDataset_SAS.csv' dlm =',' dsd firstobs=2;
length ID $8 Name $50 Type $20 Price $8 ReviewsCount $8 Rating $8 City $30 State $30;
input ID Name $ Type $ Price ReviewsCount Rating City $ State $;
run;

* Vizualizarea informațiilor referitoare la setul de date;
title "Descrierea datelor:";
proc contents data=hoteluri;
run;
```

REZULTATELE OBTINUTE:

The screenshot shows the SAS Studio interface with the 'RESULTS' tab selected. The table is titled 'WORK.HOTELURI'. The columns are labeled 'ID', 'Name', 'Type', 'Price', and 'ReviewsCount'. The data consists of 525 rows of hotel information from Amsterdam, including names like 'BUNK Hotel Amsterdam', 'YOTEL Amsterdam', and 'Multatuli Hotel', along with their room types, prices, and review counts.

ID	Name	Type	Price	ReviewsCount
1	0 BUNK Hotel Amsterdam	Bunk Pod for 2	86.76	778
2	1 YOTEL Amsterdam	Premium Double Room	167.94	500
3	2 Multatuli Hotel	Double Room	143.69	1605
4	3 nhow Amsterdam Rai	nhow Double or Twin	141.39	500
5	4 Motel One Amsterdam	Double Room	104.18	500
6	5 INNISIDE by Melia Amsterdam	The Innside Guestroom	155.35	1264
7	6 Eden Hotel Amsterdam	Small Double Room	220.66	500
8	7 citizenM Amsterdam South	King Room	156.27	500
9	8 The Alfred Hotel	Standard Double Room	139.21	2069
10	9 Andaz Amsterdam Prinsengracht - a concept by Hyatt	Observatory King	495.42	1140
11	10 Hyatt Regency Amsterdam	Twin Room	343.51	500
12	11 NH Collection Amsterdam Flower Market	Superior Double or T	323.3	500
13	12 ibis Styles Amsterdam Central Station	Small Double Room	195.78	500
14	13 Leonardo Royal Hotel Amsterdam	Queen Room - Disabl	124.7	500
15	14 Rho Hotel	Standard Double or T	206.56	2974
16	15 Leonardo Boutique Museumhotel	Double or Twin Room	168.93	500

The screenshot displays two tables from the SAS software interface:

Descrierea datelor:

Data Set Name		WORK.HOTELURI	Observations	525
Member Type		DATA	Variables	8
Engine		V9	Indexes	0
Created		05/31/2025 18:14:31	Observation Length	168
Last Modified		05/31/2025 18:14:31	Deleted Observations	0
Protection			Compressed	NO
Data Set Type			Sorted	NO
Labels				
Data Representation		SOLARIS_X86_64_LINUX_X86_64_ALPHA_TRU64_LINUX_JA64		
Encoding		utf-8 Unicode (UTF-8)		

Alphabetic List of Variables and Attributes

#	Variable	Type	Len
7	City	Char	30
1	ID	Num	8
2	Name	Char	50
4	Price	Num	8
6	Rating	Num	8
5	ReviewsCount	Num	8
8	State	Char	30
3	Type	Char	20

❖ Interpretarea rezultatelor

Rezultatele evidențiază setul de date creat, împreună cu descrierea sa. Acesta cuprinde 525 de observații și 8 variabile, fiind specificate atât tipurile de date, cât și lungimile aferente fiecărei variabile.

2. Crearea și folosirea de etichete și formate definite de utilizator

❖ Descrierea problemei

In analiza datelor, o problemă frecventă este interpretarea dificilă a valorilor brute din setul de date, mai ales când acestea sunt codificate numeric sau când denumirile variabilelor nu sunt suficient de explicite. Pentru a facilita înțelegerea și prezentarea rezultatelor, se impune utilizarea de etichete (labels) și formate definite de utilizator (user-defined formats).

Etichetele oferă descrieri mai clare și mai intuitive pentru variabile, înlocuind denumirile tehnice cu unele mai usor de înțeles. Formatele permit transformarea valorilor codificate în categorii semnificative, crescând lizibilitatea și relevanța analizelor statistice.

Astfel, aceste instrumente sunt esențiale pentru a oferi o interpretare corectă și eficientă a datelor, contribuind la o analiză mai clară și la o comunicare mai eficientă a rezultatelor.

❖ Informatii necesare pentru rezolvare

Pentru a realiza această etapă a analizei, sunt necesare următoarele informații:

- Structura variabilelor din setul de date – este esențial să cunoaștem valorile posibile ale variabilelor pentru a putea crea categorii relevante;
- Semnificația valorilor numerice – trebuie înțeles ce reprezintă scorul de rating și prețul, astfel încât să putem grupa valorile în etichete semnificative din punct de vedere analitic;

❖ Produs software/functie/metode de calcul folosite

- **Produs software folosit:** SAS Studio
- **Limbaj:** SAS
- **Functii/Metode utilizate:**
 - proc format – definește formate personalizate pentru variabile;

- value – specifică denumirile categoriilor pentru valorile continue
- data + set – creează un nou set de date (hoteluri_formatat) pe baza setului original;
- label – atribuie denumiri descriptive variabilelor;
- format – aplică formatele personalizate asupra variabilelor;
- proc print cu opțiunea (obs = 10) – afisează primele 10 observații din setul de date formatat;
- proc freq – procedura care calculează frecvența de apariție a valorilor unei variabile;
- tables – specifică variabila analizată în analiza de frecvență;
- nocom și nopercent – opțiuni care elimină coloanele de cumul și procente din tabelul de frecvență;

❖ Rezolvarea cu ajutorului produsului software

Am rezolvat această etapă definind și aplicând formate personalizate pentru variabilele numerice Rating și Price, utilizând instrucțiunea proc format, pentru a le grupa în categorii ușor de interpretat. Ulterior, am creat un nou set de date (hoteluri_formatat) în care am adăugat etichete descriptive variabilelor prin label și am aplicat formatele definite. În final, am folosit procedurile proc print și proc freq pentru a vizualiza primele observații din setul de date și pentru a analiza frecvențele valorilor grupate ale ratingului și prețului.

CODUL UTILIZAT:

```
*Crearea și folosirea de etichete și formate;
proc format;
value ratingfmt
    low <- 7      = "Slab"
    7 -< 8       = "Acceptabil"
    8 -< 9       = "Foarte bun"
    9 -> high   = "Excelent";
value pricefmt
    low <- 150   = "Economie"
    150 -< 300   = "Standard"
    300 - high   = "Premium";
run;

data hoteluri_formatat;
  set hoteluri;
  label
    ID = "Cod Hotel"
    Name = "Numele Hotelului"
    Type = "Tip Cameră"
    Price = "Preț per Noapte (EUR)"
    ReviewsCount = "Număr Recenzii"
    Rating = "Scor Rating"
    City = "Cartier"
    State = "Oraș";
  format rating ratingfmt.
    price pricefmt.
run;
```



```
*Afisarea setului de date formatat;
proc print data=hoteluri_formatat(obs=10) LABEL;
run;

*Frecvența de apariție pentru pret;
title "Frecvența de apariție pentru pret";
proc FREQ data=hoteluri_formatat;
  TABLES Price /nocom nopercent;
  FORMAT pret pricefmt.;
run;

*Frecvența de apariție pentru rating;
title "Frecvența de apariție pentru rating";
proc FREQ data=hoteluri_formatat;
  TABLES Rating /nocom nopercent;
  FORMAT rating ratingfmt.;
run;
```

REZULTATELE OBTINUTE:

Obs	Cod Hotel	Numele Hotelului	Tip Cameră	Preț per Noapte (EUR)	Număr Recenzi	Scor Rating	Cartier	Oraș
1	0	BUNK Hotel Amsterdam	Bunk Pod for 2	Economie	778	Foarte bun	Amsterdam Noord	Amsterdam
2	1	YOTEL Amsterdam	Premium Double Room	Standard	500	Foarte bun	Amsterdam Noord	Amsterdam
3		Multatul Hotel	Double Room	Economie	1605	Acceptabil	Amsterdam City Center	Amsterdam
4	3	rhow Amsterdam Rai	rhow Double or Twin	Economie	500	Excelent	Zuidamstel	Amsterdam
5	4	Motel One Amsterdam	Double Room	Economie	500	Foarte bun	Zuidamstel	Amsterdam
6	5	INNSIDE by Melia Amsterdam	The Innside Guestroo	Standard	1264	Foarte bun	Zuidamstel	Amsterdam
7	6	Eden Hotel Amsterdam	Small Double Room	Standard	500	Foarte bun	Amsterdam City Center	Amsterdam
8	7	citizenM Amsterdam South	King Room	Standard	500	Foarte bun	Zuidamstel	Amsterdam
9	8	The Alfred Hotel	Standard Double Room	Economie	2069	Acceptabil	Oud Zuid	Amsterdam
10	9	Andaz Amsterdam Prinsengracht - a concept by Hyatt	Observatory King	Premium	1140	Foarte bun	Amsterdam City Center	Amsterdam

Frecvența de apariție pentru pret

The FREQ Procedure

Preț per Noapte (EUR)	
Price	Frequency
Economie	369
Standard	130
Premium	26

Frecvența de apariție pentru rating

The FREQ Procedure

Scor Rating	
Rating	Frequency
Slab	21
Acceptabil	94
Foarte bun	317
Excelent	93

❖ Interpretarea rezultatelor

În primul rând, am afișat primele 10 observații din setul de date formatat, evidențiind noile etichete atribuite variabilelor și categoriile aferente pentru preț și rating, ceea ce a făcut datele mai clare și mai ușor de interpretat. Ulterior, am afișat frecvențele de apariție pentru categoriile de preț și rating. În ceea ce privește prețurile, observăm că majoritatea hotelurilor (369) se încadrează în categoria „Economie”, urmate de 130 în categoria „Standard” și doar 26 în categoria „Premium”. Pentru rating, cele mai multe hoteluri sunt clasificate ca „Foarte bune” (317), urmate de un număr aproximativ egal de hoteluri „Acceptabile” (94) și „Excelente” (93), în timp ce doar 21 sunt considerate „Slabe”. Acest lucru sugerează că piața hotelieră analizată este dominată de opțiuni accesibile și servicii de calitate medie spre înaltă, ceea ce poate indica o concurență ridicată în segmentul economic și o bună satisfacție a clienților în general.

3. Procesarea condițională a datelor

❖ Descrierea problemei

Problema vizată prin utilizarea procesării condiționale a datelor este extragerea și analizarea segmentelor relevante dintr-un set de date complex, în funcție de anumite criterii specifice. Scopul acestei abordări este de a filtra informațiile pentru a evidenția doar hotelurile care respectă anumite condiții – precum locația, tipul camerei, numărul de recenzii, prețul sau ratingul – și de a facilita astfel interpretarea direcționată a datelor.

Necesitatea procesării condiționale apare din dorința de a face analize personalizate pe categorii de interes, de exemplu identificarea hotelurilor populare, a celor cu servicii bune, sau a celor încadrate într-o anumită gamă de preț, oferind o imagine clară pentru luarea deciziilor strategice.

❖ Informatii necesare pentru rezolvare

- Numele variabilelor din setul de date;
- Intervalele valorice pentru realizarea clasificărilor;
- Criteriile de filtrare relevante pentru analiza: tipurile de camere vizate, numarul minim de recenzii, localizarea geografica a hotelurilor, etc.

❖ Produs software/functie/metode de calcul folosite

- **Produs software folosit:** SAS Studio
- **Limbaj:** SAS
- **Functii/Metode utilizate:**
 - proc print — pentru afișarea datelor filtrate și a subseturilor de date;
 - data + set - pentru crearea unui subset de date nou;
 - instrucțiunea where — pentru filtrarea observațiilor pe baza unor condiții specifice;
 - expresii condiționale if-else — pentru clasificarea datelor în categorii;

❖ Rezolvarea cu ajutorului produsului software

Procesarea condițională a datelor a fost realizată în mai mulți pași pentru a extrage și analiza informații relevante despre hoteluri. Mai întâi, am filtrat hotelurile din Amsterdam Noord

care au un rating peste 8 sau un preț mai mare de 200 EUR, folosind instrucțiunea WHERE în proc print și afișând doar variabilele esențiale precum numele, orașul, ratingul și prețul. Apoi, am creat un subset numit „hoteluri_populare”, utilizând un pas de date (data step) împreună cu instrucțiunea WHERE, care include doar camere de tip „Double Room” sau „King Room” cu cel puțin 500 de recenzii, pentru a analiza mai atent segmentul de hoteluri foarte apreciate. Ulterior, am selectat hotelurile din Nijmegen cu prețuri medii între 150 și 300 EUR și ratinguri foarte bune, peste 8.5, aplicând din nou filtrarea condiționată cu WHERE în proc print pentru o perspectivă mai specifică pe această piață. În final, am realizat o clasificare a hotelurilor în funcție de preț, atribuindu-le etichete descriptive — „Mic”, „Mediu” sau „Mare” — prin utilizarea unei structuri condiționale IF-ELSE IF în cadrul unui pas de date, pentru a înțelege mai bine distribuția prețurilor și a facilita interpretarea datelor. Astfel, am combinat filtrarea condiționată, crearea de subseturi și clasificarea condiționată pentru o analiză clară și structurată a datasetului.

CODUL UTILIZAT:

```
*Procesare conditională;
* WHERE;
* Ex 1;
title "Hoteluri din Amsterdam Noord cu rating peste 8 sau pret peste 200 EUR";
proc print data=hoteluri;
  where City = 'Amsterdam Noord' and
        (Rating > 8 or Price > 200);
  var Name City Rating Price;
run;

*Ex 2 - crearea de subset și folosirea where;
data hoteluri_populare;
  set hoteluri;
  where (Type in ('Double Room', 'King Room')) and ReviewsCount ge 500;
run;

title "Camere Double sau King Room, cu minim 500 recenzii";
proc print data = hoteluri_populare;
  var Name City Price Rating;
run;

*Ex 3;
title "Hoteluri din Nijmegen cu pret mediu și rating foarte bun";
proc print data=hoteluri;
  where City = 'Nijmegen' and
        Price between 150 and 300 and Rating gt 8.5;
  var Name City Price Rating;
run;

* Clasificare prețuri în grupe;
data hoteluri_clasificat;
length PriceCategory $10;
  set hoteluri;
  if Price < 100 and not missing(Price) then PriceCategory = "Mic";
  else if 100 <= Price < 200 then PriceCategory = "Mediu";
  else if Price >= 200 then PriceCategory = "Mare";
run;

title "Hoteluri cu categorii de pret";
proc print data=hoteluri_clasificat;
  var Name City Price PriceCategory;
run;
```

REZULTATELE OBTINUTE:

Hoteluri din Amsterdam Noord cu rating peste 8 sau pret peste 200 EUR

Obs	Name	City	Rating	Price
13	BUNK Hotel Amsterdam	Amsterdam Noord	8.4	86.76
14	YOTEL Amsterdam	Amsterdam Noord	8.1	167.94

Camere Double sau King Room, cu minim 500 recenzii				
Obs	Name	City	Price	Rating
1	Multatuli Hotel	Amsterdam City Center	143.69	7.4
2	Holiday Inn Express Amsterdam - City Hall, an IHG	Amsterdam City Center	190.39	8.4
3	Hotel Sutor	Breda	91.16	8.6
4	Camarille Hotel & Restaurant Breda	Breda	67.42	7.2
5	Premiere Classe Hotel Breda	Breda	43.87	6.5
6	Savoy Hotel Rotterdam	Centrum	95.77	8.5
7	citizenM Rotterdam	Centrum	124.83	8.8
8	Hotel de Koophandel	Delft	107.77	8.7
9	Shanghai Hotel Holland	Delft	77.25	7.6
10	Camarille Hotel & Restaurant Delft	Delft	60.67	7.1

Hoteluri din Nijmegen cu pret mediu și rating foarte bun

Obs	Name	City	Price	Rating
293	Boutique Hotel Strelman	Nijmegen	151.78	9.4
294	B&B Stads oase Nijmegen	Nijmegen	269.42	9.3

Hoteluri cu categorii de pret

Obs	Name	City	Price	PriceCategory
1	Multatuli Hotel	Amsterdam City Center	143.69	Mediu
2	Eden Hotel Amsterdam	Amsterdam City Center	220.66	Mare
3	Andaz Amsterdam Prinsengracht - a concept by Hyatt	Amsterdam City Center	495.42	Mare
4	Hyatt Regency Amsterdam	Amsterdam City Center	343.51	Mare
5	NH Collection Amsterdam Flower Market	Amsterdam City Center	323.30	Mare
6	ibis Styles Amsterdam Central Station	Amsterdam City Center	195.78	Mediu
7	Rho Hotel	Amsterdam City Center	206.56	Mare
8	Hotel V Nesplein	Amsterdam City Center	241.09	Mare
9	Kimpton De Witt Amsterdam, an IHG Hotel	Amsterdam City Center	380.79	Mare
10	The White Tulip Hostel	Amsterdam City Center	134.71	Mediu

❖ Interpretarea rezultatelor

După ce am filtrat hotelurile cu rating peste 8 sau preț peste 200 EUR, am identificat doar două unități, ambele cu rating peste 8, dar cu prețuri sub 200 EUR. Acest lucru indică faptul că în Amsterdam Noord predomină hotelurile accesibile, de tip economy, sugerând o piață orientată către turiști care preferă cazarea la prețuri rezonabile, fără multe opțiuni de lux sau premium.

În urma filtrării hotelurilor populare ce oferă camere Double Room sau King Room și au cel puțin 500 de recenzii, am obținut 40 de observații, evidențiind existența unui segment puternic de hoteluri bine evaluate și frecventate, ceea ce reflectă încrederea și reputația pozitivă pe piață.

Pentru hotelurile din Nijmegen, selecția celor cu prețuri între 150 și 300 EUR și ratinguri peste 8.5 a dus la identificarea a doar două hoteluri care respectă aceste criterii, semnalând o ofertă mai restrânsă de unități de nivel mediu spre premium și de calitate ridicată, ceea ce poate reprezenta o oportunitate pentru extinderea pe piață.

În final, clasificarea hotelurilor pe categorii de preț ne-a oferit o distribuție clară pe diferite zone, facilitând o analiză mai structurată și o mai bună înțelegere a segmentării pieței în funcție de preț.

4. Procesarea iterativa a datelor

❖ Descrierea problemei

Procesarea iterativă este utilizată pentru a calcula valori cumulate sau rezultate determinate prin repetarea unor condiții pentru fiecare observație. În acest caz, o vom utiliza pentru a calcula un scor total al hotelurilor, combinând punctaje din mai multe variabile prin adunări succesive condiționate, și pentru a determina numărul maxim de zile de cazare într-un buget fix, printr-o buclă care scade prețul noptilor din buget până la epuizare. Această abordare iterativă este necesară pentru a putea acumula punctaje pe baza condițiilor variabile și pentru a

simula consumul bugetului în timp, oferind o analiză mai dinamică și detaliată a datelor despre hoteluri și opțiunile de cazare.

❖ ***Informatii necesare pentru rezolvare***

- Datele de intrare, reprezentate de variabilele din setul de date initial;
- Bugetul fix definit pentru calculul zilelor de cazare: 500 EUR
- Condițiile pentru atribuirea punctajelor în calculul scorului total: intervalele valorilor pentru Rating ($\geq 9, \geq 8, \geq 7$), intervalele valorilor pentru ReviewsCount ($\geq 1000, \geq 500$), pragul pentru Price (< 150)
- Condiția pentru iterarea zilelor în funcție de preț și buget: bugetul trebuie să acopere prețul nopții, iar prețul să nu lipsească;
- Categoriile predefinite pentru clasificarea duratei sejurului în funcție de numărul de zile: ≥ 5 zile: „Sejur lung”; ≥ 3 zile: „Sejur mediu”; > 0 zile: „Sejur scurt”; altfel „Buget insuficient”.

❖ ***Produs software/functie/metode de calcul folosite***

- **Produs software folosit:** SAS Studio
- **Limbaj:** SAS
- **Functii/Metode utilizate:**
 - data — pentru crearea unui nou set de date;
 - set — pentru citirea observațiilor dintr-un set de date existent;
 - length — pentru definirea lungimii și tipului unei variabile noi;
 - Atribuire și actualizare variabile
 - Instrucțiuni conditionale if... else if... else — pentru aplicarea regulilor de scor și clasificare în funcție de valori;
 - do while — pentru realizarea unei bucle iterative care se repetă cât timp o condiție este adeverată (calculul numărului de zile în limita bugetului);
 - Operatorul aritmetic de scădere (-=) — pentru ajustarea bugetului la fiecare iterare;
 - proc print — pentru afișarea unui subset de date sau a unor variabile specifice;

❖ ***Rezolvarea cu ajutorul produsului software***

Mai întâi, s-a creat un nou set de date în care pentru fiecare hotel din lista inițială s-a calculat un scor total pe baza mai multor criterii: ratingul, numărul de recenzii și prețul. Astfel, pentru fiecare hotel, scorul a fost acumulat incremental, acordându-se puncte suplimentare în funcție de intervalele de rating (de exemplu, mai mult de 9 primește 3 puncte), numărul de recenzii (peste 1000 recenzii primește 2 puncte) și prețul sub 150 EUR (primește 1 punct). În cadrul urmatorului exemplu, s-a calculat numărul maxim de zile de cazare posibile în limita unui buget fix de 500 EUR. Această etapă a folosit o buclă iterativă „do while” care a scăzut succesiv prețul unei nopți din buget și a numărat zilele până când bugetul nu a mai permis o nouă noapte. La final, pentru fiecare hotel, durata posibilă a sejurului a fost clasificată în „sejur lung”, „sejur mediu”, „sejur scurt” sau „buget insuficient” pe baza numărului de zile calculate. Astfel este prezentata procesarea iterativa in cadrul a doua exemple: calculul condiționat al scorului și

procesarea iterativă pentru simularea consumului bugetului, oferind o analiză detaliată și dinamică a opțiunilor de cazare.

CODUL UTILIZAT:

```
*Procesare iterativa;
*Determinarea scorului total pe baza mai multor variabile;
data scoruri_hoteluri;
  set hoteluri;

  length ScorTotal 8;
  ScorTotal = 0;

  if Rating >= 9 then ScorTotal + 3;
  else if Rating >= 8 then ScorTotal + 2;
  else if Rating >= 7 then ScorTotal + 1;

  if ReviewsCount >= 1000 then ScorTotal + 2;
  else if ReviewsCount >= 500 then ScorTotal + 1;
  else if Price < 150 then ScorTotal + 1;

run;

proc print data=scoruri_hoteluri(obs=10);
  var Name Rating ReviewsCount Price ScorTotal;
run;

*Calcularea nr de zile pana la depasirea bugetului de 500 EUR;
data sejur;
  set hoteluri;
  length Sejur $ 20;
  buget = 500;
  Zile = 0;

  do while (buget >= Price and not missing(Price));
    buget = buget - Price;
    zile + 1;
  end;

  if zile >= 5 then Sejur = "Sejur lung";
  else if zile >= 3 then Sejur = "Sejur mediu";
  else if zile > 0 then Sejur = "Sejur scurt";
  else Sejur = "Buget insuficient";
run;

title "Număr de zile posibile la hoteluri din bugetul de 500 EUR";
proc print data=sejur;
  var Name City Price Zile Sejur;
run;
```

REZULTATELE OBTINUTE:

Obs	Name	Rating	ReviewsCount	Price	ScorTotal
1	BUNK Hotel Amsterdam	8.4	778	86.76	4
2	YOTEL Amsterdam	8.1	500	167.94	3
3	Multatuli Hotel	7.4	1605	143.69	4
4	nhow Amsterdam Rai	9.0	500	141.39	5
5	Motel One Amsterdam	8.8	500	104.18	4
6	INNSIDE by Meliá Amsterdam	8.4	1264	155.35	4
7	Eden Hotel Amsterdam	8.3	500	220.66	3
8	citizenM Amsterdam South	8.8	500	156.27	3
9	The Alfred Hotel	7.3	2069	139.21	4
10	Andaz Amsterdam Prinsengracht - a concept by Hyatt	8.9	1140	495.42	4

Număr de zile posibile la hoteluri din bugetul de 500 EUR

Obs	Name	City	Price	Zile	Sejur
1	BUNK Hotel Amsterdam	Amsterdam Noord	86.76	5	Sejur lung
2	YOTEL Amsterdam	Amsterdam Noord	167.94	2	Sejur scurt
3	Multatuli Hotel	Amsterdam City Center	143.69	3	Sejur mediu
4	nhow Amsterdam Rai	Zuidamstel	141.39	3	Sejur mediu
5	Motel One Amsterdam	Zuidamstel	104.18	4	Sejur mediu
6	INNSIDE by Meliá Amsterdam	Zuidamstel	155.35	3	Sejur mediu
7	Eden Hotel Amsterdam	Amsterdam City Center	220.66	2	Sejur scurt
8	citizenM Amsterdam South	Zuidamstel	156.27	3	Sejur mediu
9	The Alfred Hotel	Oud Zuid	139.21	3	Sejur mediu
10	Andaz Amsterdam Prinsengracht - a concept by Hyatt	Amsterdam City Center	495.42	1	Sejur scurt
11	Hyatt Regency Amsterdam	Amsterdam City Center	343.51	1	Sejur scurt
12	NH Collection Amsterdam Flower Market	Amsterdam City Center	323.30	1	Sejur scurt
13	Ibis Styles Amsterdam Central Station	Amsterdam City Center	195.78	2	Sejur scurt
14	Leonardo Royal Hotel Amsterdam	Oost	124.70	4	Sejur mediu
15	Rho Hotel	Amsterdam City Center	206.56	2	Sejur scurt
16	Leonardo Boutique Museumhotel	Oud Zuid	168.93	2	Sejur scurt
17	Hotel V Nesplein	Amsterdam City Center	241.09	2	Sejur scurt
18	Kimpton De Witt Amsterdam, an IHG Hotel	Amsterdam City Center	380.79	1	Sejur scurt
19	Qbic Hotel WTC Amsterdam	Zuidamstel	88.91	5	Sejur lung
20	Hotel Espresso	Oud West	142.91	3	Sejur mediu
21	Hotel Central Park	Oud Zuid	74.73	6	Sejur lung
22	The White Tulip Hostel	Amsterdam City Center	134.71	3	Sejur mediu
23	Swissotel Amsterdam	Amsterdam City Center	233.15	2	Sejur scurt

❖ Interpretarea rezultatelor

În primul output, primele 10 observații arată că majoritatea hotelurilor au un scor total de 4, iar doar unul are scorul 5. Aceasta indică faptul că majoritatea hotelurilor îndeplinește în mod similar criteriile evaluate — rating bun, număr moderat de recenzii și preț accesibil — dar există puține hoteluri care excelează în toate aceste aspecte concomitent.

În al doilea output, sunt afișate numărul de zile pe care le poate petrece o persoană la fiecare hotel în limita unui buget de 500 EUR, împreună cu clasificarea tipului de sejur (scurt, mediu sau lung). Aceste rezultate evidențiază diferențele semnificative de preț între hoteluri; de exemplu, primul hotel permite un sejur lung de 5 zile, în timp ce al 10-lea hotel poate fi rezervat doar pentru o zi. Această variație reflectă diferențele mari de prețuri, care pot fi influențate de factori precum locația hotelului, nivelul de confort oferit sau popularitatea acestuia. Astfel, analiza oferă o perspectivă clară asupra raportului calitate-preț și a opțiunilor de cazare în funcție de buget.

5. Utilizarea functiilor SAS

❖ *Descrierea problemei*

În continuare, am utilizat funcțiile SAS pentru a facilita prelucrarea și analiza datelor. Funcțiile text permit manipularea datelor textuale, esențiale pentru a asigura coerență și comparabilitatea valorilor din setul de date. Funcțiile SQL sunt folosite pentru a adăuga și să sintetizeze datele, oferind statistică descriptivă care ajută la identificarea tendințelor și a modelelor în cadrul datelor. Scopul acestor funcții este de a eficientiza procesul de analiză și de a sprijini luarea deciziilor bazate pe date corecte și bine structurate.

❖ *Informatii necesare pentru rezolvare*

- Datele inițiale cu informații despre hoteluri, în special câmpurile textuale precum City și Name, și variabile numerice precum Price și Rating;

❖ *Produs software/functie/metode de calcul folosite*

- **Produs software folosit:** SAS Studio
- **Limbaj:** SAS
- **Functii/Metode utilizate:**

- lowercase() – convertește textul în litere mici;
- uppercase() – convertește textul în litere mari;
- scan() – extrage un cuvânt specific dintr-un sir de caractere;
- substr() – extrage un sir de caractere dintr-un text, începând de la o poziție dată și pe o anumită lungime;
- length() – returnează lungimea unui sir de caractere;
- count() – numără observațiile din fiecare grup;
- mean() – calculează media valorilor numerice pe grupuri;
- max() – identifică valoarea maximă pe grupuri;
- group by – grupează datele după o variabilă;
- proc print;
- proc sql pentru interogări și agregări pe setul de date.

❖ *Rezolvarea cu ajutorul produsului software*

Mai întâi, prin intermediul unui data step, se prelucrează textul din variabilele existente în setul de date hoteluri. Pentru fiecare observație, se creează noi variabile folosind funcții specifice: lowercase() transformă numele orașului în litere mici, uppercase() transformă

numele hotelului în litere mari, scan() extrage primul cuvânt din numele hotelului, substr() ia primele trei caractere din numele hotelului, iar length() calculează lungimea totală a numelui hotelului. Aceste transformări facilitează analizarea și standardizarea textului. Apoi, cu ajutorul procedurii proc print, se afișează un tabel care conține variabilele originale și cele noi create, pentru a verifica corectitudinea și relevanța transformărilor. Ulterior, în procedura proc sql, se aplică funcții agregate pe setul original de date hoteluri: se grupează hotelurile după oraș (City) și se calculează numărul total de hoteluri (count()), prețul mediu (mean()) și cel mai mare rating (max()) pentru fiecare oraș. Această etapă oferă o sinteză a datelor, evidențiind caracteristicile principale ale hotelurilor pe orașe. Astfel, procesul combină prelucrarea detaliată a textului cu analiza agregată, pentru a obține informații relevante și bine structurate despre hoteluri.

CODUL UTILIZAT:

```
*Utilizarea functiilor SAS;
* Utilizarea functiilor text;
data hoteluri_text;
  set hoteluri;
  city_lower = lowercase(City);
  name_upper = uppercase(Name);
  first_word = scan(name, 1);
  initials = substr(name, 1, 3);
  length_name = length(name);
run;

proc print data = hoteluri_text;
  var City Name City_lower Name_upper First_word Initials Length_Name;
run;

proc sql;
  select City,
    count(*) as NrHoteluri,
    mean(Price) as PretMediu format=8.2,
    max(Rating) as MaxRating
  from hoteluri
  group by City;
quit;
```

REZULTATELE OBTINUTE:

Obs	City	Name	City_lower	Name_upper	First_word	Initials	Length_Name
1	Amsterdam Noord	BUNK Hotel Amsterdam	amsterdam noord	BUNK HOTEL AMSTERDAM	BUNK	BUN	20
2	Amsterdam Noord	YOTEL Amsterdam	amsterdam noord	YOTELAMSTERDAM	YOTEL	YOT	15
3	Amsterdam City Center	Multatuli Hotel	amsterdam city center	MULTATULI HOTEL	Multatuli	Mul	15
4	Zuideramstel	nhow Amsterdam Rai	zuideramstel	NHOW AMSTERDAM RAI	nhow	nho	18
5	Zuideramstel	Motel One Amsterdam	zuideramstel	MOTEL ONE AMSTERDAM	Motel	Mot	19
6	Zuideramstel	INNSIDE by Meliá Amsterdam	zuideramstel	INNSIDE BY MELÍA AMSTERDAM	INNSIDE	INN	27
7	Amsterdam City Center	Eden Hotel Amsterdam	amsterdam city center	EDEN HOTEL AMSTERDAM	Eden	Ede	20
8	Zuideramstel	citizenM Amsterdam South	zuideramstel	CITIZENM AMSTERDAM SOUTH	citizenM	cit	24
9	Oud Zuid	The Alfred Hotel	oud zuid	THE ALFRED HOTEL	The	The	16
10	Amsterdam City Center	Andaz Amsterdam Prinsengracht - a concept by Hyatt	amsterdam city center	ANDAZ AMSTERDAM PRINSENGRACHT - A CONCEPT BY HYATT	Andaz	And	50

City	NrHoteluri	PretMediu	MaxRating
Amsterdam City Center	12	259.09	8.9
Amsterdam Noord	2	127.35	8.4
Bemelen	25	250.27	10
Boschstraatkwartier	2	317.19	8
Breda	20	103.99	9.4
Centrum	14	121.77	9.1
City Centre	19	137.62	9.4
Delfshaven	1	74.73	9.1
Delft	20	124.49	9.2
Den Bosch	23	123.72	9.4
Eindhoven	1	68.26	8.1

❖ *Interpretarea rezultatelor*

Tabelul prezintă rezultatele aplicării unor funcții text pe un set de date cu hoteluri din Amsterdam. Pentru fiecare hotel, se arată valorile originale și cele prelucrate: „City_lower” conține orașul în litere mici pentru uniformizare, „Name_upper” transformă numele în litere mari pentru o comparare ușoară, „First_word” extrage primul cuvânt din nume, „Initials” reține primele trei caractere pentru coduri scurte, iar „Length_Name” indică lungimea numelui hotelului. Aceste transformări evidențiază cum funcțiile text din SAS ajută la standardizarea și analiza eficientă a datelor.

Tabelul prezintă rezultatele unei interogări SQL care grupează hotelurile după oraș și calculează numărul de hoteluri, prețul mediu și cel mai mare rating pentru fiecare oraș. Orașe ca Bemelen, Haarlem sau Den Bosch au multe hoteluri (25), în timp ce altele precum Haagse Hout sau Kralingen-Crooswijk au puține. Prețurile medii sunt mai mari în zone turistice centrale (ex. Boschstraatkwartier, Amsterdam City Center) și mai mici în orașe ca Eindhoven sau Breda. Majoritatea orașelor au hoteluri cu ratinguri ridicate, unele chiar de 10, indicând servicii de calitate. Datele reflectă astfel diversitatea ofertei hoteliere în număr, preț și calitate.

6. Combinarea seturilor de date prin proceduri specifice SAS și SQL

❖ *Descrierea problemei*

Scopul problemei vizează identificarea hotelurilor care au atât un preț accesibil, sub 200 EUR pe noapte, cât și un rating foarte bun, egal sau peste 8.5. Această combinație ajută la selecția celor mai bune opțiuni de cazare, oferind un bun raport calitate-preț pentru clienți.

❖ *Informații necesare pentru rezolvare*

- Setul de date complet „hoteluri”, care conține variabilele: ID (identificator unic), Name (numele hotelului), Price (preț pe noapte), Rating (scorul acordat de utilizatori), City, State.
- Criteriile de filtrare: Price < 200 și Rating >= 8.5.
- Necesitatea combinării a două subseturi rezultate din aplicarea acestor criterii.

❖ *Produs software / funcție / metodă de calcul folosită*

- **Software:** SAS

- **Metode folosite:**

-Procedura DATA step pentru filtrare și subsetare (crearea subseturilor „hoteluri_ieftine” și „hoteluri_bune”)

- Procedura DATA step cu comanda MERGE pentru combinarea subseturilor pe baza ID-ului

-PROC SQL cu INNER JOIN pentru combinarea subseturilor prin intersecție.

-Se asigură sortarea datelor după ID înainte de combinare, pentru a respecta cerințele comenzii MERGE.

❖ Rezolvarea cu ajutorul produsului software

Au fost create două subseturi prin filtrarea datasetului inițial pe baza a două criterii: prețul și ratingul hotelurilor. Ulterior, aceste subseturi au fost sortate după variabila ID pentru a facilita combinarea lor. În primul caz, s-a utilizat un DATA step cu instrucțiunea MERGE, păstrând doar hotelurile comune celor două subseturi. În al doilea caz, s-a realizat un INNER JOIN folosind PROC SQL, având ca bază ID-urile comune. Ambele metode au condus la obținerea unor tabele ce conțin hoteluri cu preț sub 200 și rating de cel puțin 8.5. Rezultatele au fost afișate cu ajutorul PROC PRINT, fiind limitate la primele 10 observații pentru a asigura claritatea afișării.

CODUL UTILIZAT:

```

data hoteluri_ieftine;
  set hoteluri;
  where Price < 200;
run;

*Subset 2: hoteluri cu rating foarte bun (>= 8.5);
data hoteluri_bune;
  set hoteluri;
  where Rating >= 8.5;
run;

proc sort data=hoteluri_ieftine; by ID; run;
proc sort data=hoteluri_bune; by ID; run;

*Varianta 1 utilizând MERGE;
data hoteluri_combinate;
  merge hoteluri_ieftine(in=a) hoteluri_bune(in=b);
  by ID;
  if a and b; * păstrează doar hotelurile care au preț sub 200 și rating >= 8.5;
run;

proc print data=hoteluri_combinate(obs=10);
  var ID Name Price Rating City State;
  title "Hoteluri cu pret sub 200 și rating >= 8.5 (DATA step merge)";
run;

*Varianta 2 utilizând PROC SQL;
proc sql;
  create table hoteluri_intersecție_sql as
  select a.*
  from hoteluri_ieftine as a
  inner join hoteluri_bune as b
  on a.ID = b.ID;
quit;

proc print data=hoteluri_intersecție_sql(obs=10);
  var ID Name Price Rating City State;
  title "Hoteluri cu pret sub 200 și rating >= 8.5 (PROC SQL inner join)";
run;

```

REZULTATELE OBTINUTE:

Hoteluri cu pret sub 200 și rating >= 8.5 (DATA step merge)

Obs	ID	Name	Price	Rating	City	State
1	100	Hotel 't Keershuys	132.24	8.9	Den Bosch	North Brabant
2	101	KASeme Boutique Hotel	155.19	8.7	Den Bosch	North Brabant
3	113	Hot, a luxury B&B in the center of Eindhoven	144.60	9.7	Strijp	Eindhoven
4	114	EindhovenYou	56.19	9.1	Strijp	Eindhoven
5	116	cafe 't Vonderke	62.84	8.7	Strijp	Eindhoven
6	117	Queen Hotel	67.40	8.8	Eindhoven City Centre	Eindhoven
7	118	Hotel Parkzicht Eindhoven	68.57	8.6	Stratum	Eindhoven
8	119	The Student Hotel Eindhoven	69.16	8.8	Eindhoven City Centre	Eindhoven
9	120	Hotel the Match	70.95	8.7	Eindhoven City Centre	Eindhoven
10	121	Design Hotel Glow	73.81	8.5	Eindhoven City Centre	Eindhoven

Hoteluri cu pret sub 200 și rating >= 8.5 (PROC SQL inner join)

Obs	ID	Name	Price	Rating	City	State
1	100	Hotel 't Keershuys	132.24	8.9	Den Bosch	North Brabant
2	101	KASeme Boutique Hotel	155.19	8.7	Den Bosch	North Brabant
3	113	Hot, a luxury B&B in the center of Eindhoven	144.60	9.7	Strijp	Eindhoven
4	114	EindhovenYou	56.19	9.1	Strijp	Eindhoven
5	116	cafe 't Vonderke	62.84	8.7	Strijp	Eindhoven
6	117	Queen Hotel	67.40	8.8	Eindhoven City Centre	Eindhoven
7	118	Hotel Parkzicht Eindhoven	68.57	8.6	Stratum	Eindhoven
8	119	The Student Hotel Eindhoven	69.16	8.8	Eindhoven City Centre	Eindhoven
9	120	Hotel the Match	70.95	8.7	Eindhoven City Centre	Eindhoven
10	121	Design Hotel Glow	73.81	8.5	Eindhoven City Centre	Eindhoven

❖ Interpretarea rezultatelor

Analiza a extras un set de hoteluri care îndeplinește condițiile de a avea un preț sub 200 EUR și un rating de cel puțin 8.5, evidențiind astfel opțiuni cu un bun raport calitate-preț. Din acestea, au fost afișate primele 10 hoteluri, majoritatea situate în zona Eindhoven și Den Bosch, în regiunea North Brabant. Prețurile variază între 56 și 155 EUR, iar rating-urile indică o satisfacție ridicată a clientilor, cu valori cuprinse între 8.5 și 9.7. Rezultatele obținute atât prin metoda DATA step MERGE, cât și prin PROC SQL INNER JOIN sunt identice, confirmând corectitudinea și consistența selecției. Aceste date pot fi utilizate pentru a recomanda turiștilor hoteluri accesibile, dar bine cotate, în zonele respective.

7. Masive

❖ Descrierea problemei

Am realizat evaluarea hotelurilor pe baza unor criterii simple, pentru a obține un scor care să indice căte condiții importante îndeplinește fiecare hotel, facilitând astfel selecția celor mai atractive opțiuni pentru turiști.

❖ *Informații necesare pentru rezolvare*

Datele necesare includ ratingul hotelului (Rating), numarul de recenzii (ReviewsCount) si pretul camerei (Price) pentru fiecare hotel. Aceste variabile vor fi utilizate pentru a verifica daca un hotel indeplineste conditiile prestabilite: rating de cel putin 8, cel putin 500 recenzii si pret mai mic de 200 EUR..

❖ *Produs software / funcție / metodă de calcul folosită*

- **Software:** SAS (DATA step cu o structură de tip array (masiv) pentru a evalua simultan condițiile).
- **Functii folosite:** Funcția sum(of conditii[*]) adună valorile logice ale condițiilor pentru a calcula un scor simplu pentru fiecare hotel.

❖ *Rezolvarea cu ajutorul produsului software*

A fost creat un nou dataset prin preluarea datelor din tabelul inițial *hoteluri*. În cadrul unui DATA step, s-a definit un array cu trei elemente, fiecare corespunzător uneia dintre cele trei condiții de evaluare: rating, număr de recenzii și preț. Pentru fiecare hotel, au fost evaluate aceste condiții — rating mai mare sau egal cu 8, număr de recenzii cel puțin 500 și preț mai mic de 200 — și s-au atribuit valori binare (1 dacă condiția este îndeplinită, 0 în caz contrar) fiecărui element din array. Ulterior, s-a calculat un scor simplu ca suma valorilor din array, reprezentând numărul total de condiții îndeplinite de fiecare hotel. Rezultatele au fost afișate utilizând PROC PRINT, afișarea fiind limitată la primele 10 hoteluri pentru o vizualizare clară și concisă.

CODUL UTILIZAT:

```
*Masive;
data hoteluri_scor_simple;
set hoteluri;
array conditii[3];
conditii[1] = (Rating >= 8);
conditii[2] = (ReviewsCount >= 500);
conditii[3] = (Price < 200);
ScorSimplu = sum(of conditii[*]);
run;

proc print data=hoteluri_scor_simple (obs=10);
var Name Rating ReviewsCount Price ScorSimplu;
title "Scor simplu: număr de condiții îndeplinite de fiecare hotel";
run;
```

REZULTATELE OBTINUTE:

Scor simplu: număr de condiții îndeplinite de fiecare hotel

Obs	Name	Rating	ReviewsCount	Price	ScorSimplu
1	BUNK Hotel Amsterdam	8.4	778	86.76	3
2	YOTEL Amsterdam	8.1	500	107.94	3
3	Hyatt Regency Amsterdam	8.5	500	343.81	2
4	Hotel 't Keershuys	8.9	1143	132.24	3
5	KASerne Boutique Hotel	8.7	64	155.19	2
6	Kloosterhotel de Soete Moeder	8.3	994	99.14	3
7	The Den, 's-Hertogenbosch, a Tribute Portfolio Hot	8.3	168	98.47	2
8	Landgoed Huize Bergen Den Bosch - Vught	8.0	500	112.62	3
9	Best Western Plus City Centre Hotel Den Bosch	8.4	500	131.14	3
10	Tussen Gracht en SintJan	8.0	158	160.75	2

❖ *Interpretarea rezultatelor*

Rezultatul obținut în urma calculului *Scorului Simplu* evidențiază câte dintre cele trei condiții importante (rating ≥ 8 , cel puțin 500 de recenzii și preț sub 200 EUR) sunt îndeplinite de fiecare hotel. Un scor de 3 indică faptul că hotelul respectiv respectă toate cele trei criterii, ceea ce îl face o opțiune echilibrată din perspectiva calității, popularității și accesibilității financiare. De exemplu, hoteluri precum „BUNK Hotel Amsterdam” sau „Hotel 't Keershuys” au obținut scorul maxim, demonstrând că sunt atât bine cotate, cât și accesibile și populare. Pe de altă parte, hoteluri precum „Hyatt Regency Amsterdam” sau „KASerne Boutique Hotel” au un scor de 2, semnalând că nu îndeplinesc una dintre condiții(cea legată de preț sau de numărul de recenzii). Astfel, acest scor simplificat permite o evaluare rapidă și comparativă între opțiuni, ajutând utilizatorii să identifice cu ușurință hotelurile care oferă cel mai bun raport între calitate, preț și popularitate.

8. Utilizarea de proceduri pentru raportare

8.1.Raport detaliat pe orase

❖ Descrierea problemei

Obiectivul acestei proceduri este generarea unui raport detaliat care să grupeze și să prezinte informații relevante despre hoteluri în funcție de oraș, astfel încât să fie ușor de analizat distribuția acestora și principalele caracteristici (preț, tip cameră, rating, etc.) în cadrul fiecărui oraș.

❖ Informații necesare pentru rezolvare

Datele provin din setul hoteluri, care conține variabile precum:

- Name - numele hotelului;
- Type - tipul camerei;
- Price - prețul camerei;
- ReviewsCount - numarul recenziilor;
- Rating - scorul dat de utilizatori;
- City - orașul în care se află hotelul;
- State - regiunea din care face parte orașul.

Este necesară și sortarea prealabilă a datasetului după oraș (City) pentru a putea grupa datele corespunzătoare în raportul final.

❖ Produs software / funcție / metodă de calcul folosită

- Software: SAS

Functii si optiuni folosite:

- PROC SORT
- PROC PRINT;
- Opțiunea BY;
- Opțiunea ID;
- Opțiunea SUM;
- Opțiunea LABEL;
- Opțiunea NOOBS.

❖ Rezolvarea cu ajutorul produsului software

S-a utilizat procedura PROC SORT pentru a sorta datele din datasetul *hoteluri* în funcție de oraș (*City*), facilitând astfel gruparea logică a observațiilor în raport. A fost utilizată instrucțiunea BY *City* pentru a grupa hotelurile în funcție de oraș. Variabila *Name* a fost declarată cu rol de identificator (*ID*), astfel încât să apară prima în listă pentru fiecare observație. Variabilele *Price* și *ReviewsCount* au fost sumarizate cu ajutorul opțiunii SUM, furnizând totaluri relevante pentru fiecare oraș. De asemenea, s-a folosit opțiunea LABEL pentru a înlocui numele variabilelor cu etichete explicite și ușor de înțeles de către utilizator.

CODUL UTILIZAT:

```

proc sort data=hoteluri;
  by City;
run;
proc print data=hoteluri noobs label;
  by City;
  id Name;
  sum Price ReviewsCount;
  var Type Price ReviewsCount Rating State;
  label
    Name = "Numele Hotelului"
    Type = "Tip Cameră"
    Price = "Preț (EUR)"
    ReviewsCount = "Număr Recenzii"
    Rating = "Rating"
    City = "Oraș"
    State = "Regiune";
  title "Raport detaliat hoteluri pe orașe";
run;

```

REZULTATELE OBTINUTE:

Raport detaliat hoteluri pe orașe					
Oras=Amsterdam City Center					
Numele Hotelului	Tip Cameră	Pret (EUR)	Număr Recenzi	Rating	Regiune
Hyatt Regency Amsterdam	Twin Room	343.51	500	8.5	Amsterdam
NH Collection Amsterdam Flower Market	Superior Double or T	323.30	500	8.6	Amsterdam
Ibis Styles Amsterdam Central Station	Small Double Room	195.78	500	7.9	Amsterdam
Rho Hotel	Standard Double or T	209.56	2974	8.0	Amsterdam
Hotel V Nesplein	Comfort Double or Tw	241.09	1278	8.0	Amsterdam
Kimpton De Witt Amsterdam, an IHG Hotel	1 Queen or King Bed	380.79	500	8.8	Amsterdam
Multatuli Hotel	Double Room	143.89	1805	7.4	Amsterdam
The White Tulip Hostel	Standard Twin Bunk w	134.71	2064	7.0	Amsterdam
Swissôtel Amsterdam	Classic Double Room	233.15	1529	8.6	Amsterdam
Holiday Inn Express Amsterdam - City Hall, an IHG	Double Room	190.39	500	8.4	Amsterdam
Eden Hotel Amsterdam	Small Double Room	220.86	500	8.3	Amsterdam
Andaz Amsterdam Prinsengracht - a concept by Hyatt	Observatory King	495.42	1140	8.9	Amsterdam
City		3109.05	13850		

Oras=Amsterdam Noord					
Numele Hotelului	Tip Cameră	Pret (EUR)	Număr Recenzi	Rating	Regiune
BUNK Hotel Amsterdam	Bunk Pod for 2	88.76	778	8.4	Amsterdam
YOTEL Amsterdam	Premium Double Room	167.94	500	8.1	Amsterdam
City		254.70	1278		

Oras=Bemelen					
Numele Hotelului	Tip Cameră	Pret (EUR)	Număr Recenzi	Rating	Regiune
Holiday Home Green Resort Mooi Bemelen-19	Holiday Home	230.80	1	10.0	Limburg
Holiday Home Green Resort Mooi Bemelen-2	Holiday Home	291.87	1	9.6	Limburg
Holiday Home Green Resort Mooi Bemelen-12	Holiday Home	230.80	2	9.0	Limburg
Holiday Home Green Resort Mooi Bemelen-31	Holiday Home	230.80	1	9.0	Limburg
Holiday Home Green Resort Mooi Bemelen-15	Holiday Home	230.80	1	9.0	Limburg
Holiday Home Green Resort Mooi Bemelen-10	Holiday Home	311.83	1	9.0	Limburg
Holiday Home Green Resort Mooi Bemelen-13	Holiday Home	311.83	1	9.0	Limburg
Holiday Home Green Resort Mooi Bemelen-16	Holiday Home	311.83	4	8.6	Limburg
Resort Mooi Bemelen	Two-Bedroom House	130.22	1155	8.1	Limburg
Holiday Home Green Resort Mooi Bemelen-6	Holiday Home	230.80	18	8.1	Limburg
Holiday Home Green Resort Mooi Bemelen-3	Holiday Home	230.80	5	8.1	Limburg

❖ Interpretarea rezultatelor

Am obținut o listă detaliată a hotelurilor, grupate pe orașe, care oferă o perspectivă comparativă asupra pieței de cazare din mai multe zone geografice. Informațiile prezentate includ: numele hotelului, tipul camerei disponibile, prețul per noapte în euro, numărul recenziilor primite, ratingul acordat și regiunea în care se află fiecare unitate de cazare.

Raportul este structurat clar, fiecare secțiune corespunzând unui oraș (de exemplu, Amsterdam City Center, Amsterdam Noord, Bemelen etc.), ceea ce permite o analiză eficientă și localizată a ofertei de servicii hoteliere. În cazul orașului Amsterdam City Center, se observă o gamă largă de hoteluri cu prețuri ce variază semnificativ – de la 130 EUR până la peste 490 EUR pe noapte – și un număr mare de recenzi, ceea ce indică o zonă turistică intens vizitată. Ratingurile se încadrează în general între 7.0 și 8.8, reflectând o calitate bună a serviciilor.

În Amsterdam Noord, raportul este mai restrâns, dar arată un preț mediu mai scăzut și o satisfacție ridicată a clienților (ratinguri peste 8). În schimb, zona Bemelen este reprezentată în special de unități de tip „Holiday Home”, cu prețuri mai constante și ratinguri foarte ridicate (8.0–10.0), dar cu un număr redus de recenzi în majoritatea cazurilor, ceea ce sugerează o ofertă mai puțin popularizată, dar cu servicii de calitate.

Acest raport este un instrument valoros atât pentru analiza comportamentului consumatorului în turism, cât și pentru luarea deciziilor strategice în domeniul ospitalității, oferind posibilitatea de a compara orașe, segmente de piață și niveluri de calitate.

8.2.Raport privind hotelurile din fiecare Stat

❖ *Descrierea problemei*

Se dorește generarea unui raport care să prezinte hotelurile disponibile în diverse Regiuni, împreună cu prețul pe noapte și scorul de rating. Scopul este de a oferi o imagine de ansamblu asupra distribuției și calității hotelurilor din diferite locații, pentru a sprijini luarea deciziilor turistice sau de business.

❖ *Informații necesare pentru rezolvare*

Se folosește setul de date *hoteluri_formata*t, care conține următoarele variabile esențiale:

- Name - numele hotelului;
- Price - pretul camerei;
- Rating - scorul dat de utilizatori;
- State - regiunea din care face parte orasul.

❖ *Produs software / funcție / metodă de calcul folosită*

- **Software:** SAS

Functii si optiuni folosite:

- PROC SORT
- PROC PRINT;
- Opțiunea BY;
- Opțiunea ID;
- Opțiunea SUM;
- Opțiunea LABEL;
- Opțiunea NOOBS.

❖ *Rezolvarea cu ajutorul produsului software*

S-a utilizat procedura PROC SORT pentru a sorta datele din datasetul *hoteluri_formata*t în funcție de regiune (*State*), facilitând astfel gruparea logică a observațiilor în cadrul raportului. A fost utilizată instrucțiunea BY State pentru a grupa hotelurile după regiune. Variabila *Name* a fost declarată cu rol de identificator (*ID*), astfel încât să fie afișată prima în listă pentru fiecare observație. Variabilele *Price* și *Rating* au fost selectate pentru afișare, deoarece oferă informații esențiale despre tarifele de cazare și nivelul de satisfacție al clienților. Opțiunea LABEL a fost utilizată pentru a înlocui denumirile tehnice ale variabilelor cu etichete explice și intuitive, precum „Preț pe noapte” și „Scor rating”. În final, raportul a fost afișat fără numere de ordine ale observațiilor (NOOBS) pentru a asigura un aspect mai curat.

CODUL UTILIZAT:

```
*Raport privind hotelurile din fiecare regiune;
proc sort data=hoteluri_formatat;
  by State;
run;

proc print data=hoteluri_formatat noobs label;
  by State;
  id Name;
  var Price Rating;
  label Name = "Hotel"
        State = "Oraș"
        Price = "Preț pe noapte"
        Rating = "Scor rating";
  title "Raport simplu: Hoteluri grupate pe State";
run;
```

REZULTATELE OBTINUTE:

Raport simplu: Hoteluri grupate pe State

Hotel	Oraș=Amsterdam	Preț pe noapte	Scor rating
BUNK Hotel Amsterdam		Economie	Foarte bun
YOTEL Amsterdam		Standard	Foarte bun
Multatuli Hotel		Economie	Acceptabil
nhow Amsterdam Rai		Economie	Excelent
Motel One Amsterdam		Economie	Foarte bun
INN SIDE by Meliá Amsterdam		Standard	Foarte bun
Eden Hotel Amsterdam		Standard	Foarte bun
citizenM Amsterdam South		Standard	Foarte bun
The Alfred Hotel		Economie	Acceptabil
Andaz Amsterdam Prinsengracht - a concept by Hyatt		Premium	Foarte bun
Hyatt Regency Amsterdam		Premium	Foarte bun
NH Collection Amsterdam Flower Market		Premium	Foarte bun
Ibis Styles Amsterdam Central Station		Standard	Acceptabil
Leonardo Royal Hotel Amsterdam		Economie	Foarte bun
Rho Hotel		Standard	Foarte bun
Leonardo Boutique Museumhotel		Standard	Acceptabil
Hotel V Nesplein		Standard	Foarte bun
Kimpton De Witt Amsterdam, an IHG Hotel		Premium	Foarte bun
Qbic Hotel WTC Amsterdam		Economie	Acceptabil
Hotel Espresso		Economie	Acceptabil
Hotel Central Park		Economie	Slab
The White Tulip Hostel		Economie	Acceptabil
Swissôtel Amsterdam		Standard	Foarte bun
Holiday Inn Express Amsterdam - City Hall, an IHG		Standard	Foarte bun
Hotel Atlantis Amsterdam		Economie	Acceptabil

Oraș=Eindhoven

Hotel	Oraș=Eindhoven	Preț pe noapte	Scor rating
Hof, a luxury B&B in the center of Eindhoven		Economie	Excelent
Eindhoven4you		Economie	Excelent
Kazerne		Standard	Excelent
cafe 't Wonderke		Economie	Foarte bun
Queen Hotel		Economie	Foarte bun

❖ Interpretarea rezultatelor

Raportul oferă o imagine generală a hotelurilor grupate pe orașe (în acest caz, Amsterdam și Eindhoven), prezentând pentru fiecare unitate de cazare nivelul de preț pe noapte (clasificat în Economie, Standard sau Premium) și scorul rating (de la Slab la Excelent), pe baza formatărilor definite anterior. Se observă că în Amsterdam predomină hotelurile din categoriile Economie și Standard, cu scoruri de rating în general Foarte bun sau Acceptabil, ceea ce indică o ofertă variată și echilibrată pentru diferite bugete, menținând totodată un nivel satisfăcător al serviciilor. Există și câteva hoteluri Premium, dar acestea rămân minoritare. În Eindhoven, deși lista este mai scurtă, toate hotelurile din raport se încadrează în categoria Economie, dar au un scor de rating Excelent, sugerând un raport calitate-preț foarte bun în acest oraș. Această clasificare ajută la identificarea rapidă a opțiunilor de cazare în funcție de buget și nivelul de satisfacție al clienților, per oraș.

9. Proceduri statistice

9.1. Analiza univariată a prețurilor

❖ Descrierea problemei

Aceasta problema are ca scop realizarea unei analize statistice univariate asupra prețurilor practicate de hotelurile incluse în setul de date hoteluri_formatat, în vederea identificării distribuției acestora, valorilor extreme și abaterilor de la normalitate. Această analiză este utilă pentru a înțelege comportamentul general al prețurilor și pentru a fundamenta decizii privind poziționarea tarifară.

❖ *Informații necesare pentru rezolvare*

- Setul de date hoteluri_formata, care conține informații despre hoteluri, inclusiv prețul pe noapte (Price) și numele hotelului (Name);
- Necesitatea de a examina distribuția variabilei Price, precum și de a verifica dacă aceasta urmează o distribuție normală.

❖ *Produs software / funcție / metodă de calcul folosită*

- **Software:** SAS

Functii si metode folosite:

- Procedura PROC UNIVARIATE pentru analiza statistică descriptivă univariată;
- Opțiunea HISTOGRAM / NORMAL pentru afișarea grafică a distribuției prețurilor și compararea cu distribuția normală;
- Opțiunea ID Name pentru a identifica observațiile relevante din punct de vedere statistic (ex. valori extreme).

❖ *Rezolvarea cu ajutorul produsului software*

A fost utilizată procedura PROC UNIVARIATE, aplicată variabilei *Price*, pentru a genera indicatori descriptivi precum media, mediana, deviația standard, valorile minime și maxime, informații privind simetria (*skewness*) și ascuțimea (*kurtosis*) distribuției, precum și o histogramă cu suprapunerea unei curbe normale. Opțiunea *nextrval=5* a fost folosită pentru a afișa cele mai relevante 5 valori extreme, în timp ce *nextraobs=0* a fost setată pentru a nu afișa observații suplimentare. De asemenea, identificatorul *Name* a fost utilizat pentru a evidenția ce hoteluri sunt asociate cu acele valori extreme, oferind astfel o interpretare mai clară și contextualizată a datelor analizate.

CODUL UTILIZAT:

```
*Proceduri statistice;
*Analiza univariată a prețurilor;
proc univariate data=hoteluri_formata nextrval=5 nextraobs=0;
  var Price;
  id Name;
  histogram Price / normal;
  title "Analiza univariată a prețurilor hotelurilor";
run;
```

REZULTATELE OBTINUTE:

Analiza univariată a prețurilor hotelurilor			
The UNIVARIATE Procedure			
Variable: Price (Preț per Noapte (EUR))			
Moments			
N	525	Sum Weights	525
Mean	141.992705	Sum Observations	74546.17
Std Deviation	73.3852338	Variance	5385.39254
Skewness	2.27001758	Kurtosis	7.39119910
Uncorrected SS	13406958	Corrected SS	2821945.69
Coeff Variation	51.6823973	Std Error Mean	3.20279418
Basic Statistical Measures			
Location		Variability	
Mean	141.9927	Std Deviation	73.38523
Median	124.7000	Variance	5385
Mode	230.8000	Range	543.96000
		Interquartile Range	61.07000
Tests for Location: Mu0=0			
Test	Statistic	p Value	
Student's t	t	44.33401	Pr > t <.0001
Sign	M	262.5	Pr >= M <.0001
Signed Rank	S	60037.5	Pr >= S <.0001
Quantiles (Definition 5)			
Level	Quantile		
100% Max	587.83		
99%	401.38		
95%	291.87		
90%	230.80		
75% Q3	157.17		
50% Median	124.70		
25% Q1	98.10		
10%	78.43		
5%	69.16		
1%	56.56		

❖ Interpretarea rezultatelor

Analiza univariată a prețurilor hotelurilor relevă informații esențiale despre distribuția acestei variabile. Media prețului pe noapte este de aproximativ 142 EUR, însă valoarea medianei, de 124.70 EUR, indică faptul că distribuția este asimetrică spre dreapta, fiind influențată de câteva prețuri foarte mari (outlieri). Acest aspect este susținut și de valoarea coeficientului de asimetrie (skewness = 2.27) și a kurtosisului (7.39), care semnalează o distribuție accentuată și concentrată în jurul mediei. De asemenea, abaterea standard este mare (73.39 EUR), ceea ce denotă o variabilitate semnificativă a prețurilor între hoteluri. Prețul minim observat este de 58.56 EUR (1% percentil), iar cel maxim ajunge la 587.83 EUR (100%), intervalul dintre quartila 1 și 3 fiind de 61.07 EUR. Testele de localizare indică faptul că prețurile sunt semnificativ diferite de zero ($p < 0.0001$), ceea ce confirmă relevanța statistică a mediei estimate. Astfel, putem concluziona că prețurile hotelurilor sunt variate, cu o tendință clară spre valori mai mari, însă cu o concentrație semnificativă în zona economică și standard.

9.2. Procedura MEANS

❖ Descrierea problemei

Aceasta procedura propune obținerea unor indicatori statistici descriptivi pentru variabilele referitoare la hoteluri – mai exact prețul pe noapte, scorul rating și numărul de recenzii – grupați în funcție de oraș. Scopul este identificarea diferențelor între orașe privind nivelul prețurilor, gradul de satisfacție al clienților și popularitatea hotelurilor

❖ *Informații necesare pentru rezolvare*

Pentru această analiză, este necesar setul de date hoteluri care conține următoarele variabile:

- City – orașul în care se află hotelul
- Price – prețul pe noapte
- Rating – scorul de satisfacție al clienților
- ReviewsCount – numărul de recenzii primite de fiecare hotel

❖ *Produs software / funcție / metodă de calcul folosită*

- **Software:** SAS

Functii si metode folosite:

- Procedura PROC MEANS, care calculează automat indicatori statistici pentru variabile numerice, precum: valoarea minimă, valoarea maximă, media, numărul de observații valide (N), numărul de observații lipsă (NMISS) și suma.

❖ *Rezolvarea cu ajutorul produsului software*

În cadrul analizei, a fost utilizată instrucțiunea class City pentru a grupa observațiile în funcție de oraș, permîțând astfel compararea indicatorilor descriptivi între diferite locații. Variabilele selectate pentru analiză au fost Price, Rating și ReviewsCount, utilizând instrucțiunea var. Pentru fiecare dintre aceste variabile, s-au calculat indicatori statistici esențiali: valoarea maximă (max), valoarea minimă (min), media (mean), numărul de observații nenule (n), numărul de valori lipsă (nmiss) și suma totală (sum), oferind o imagine detaliată a distribuției și completitudinii datelor pentru fiecare oraș analizat.

CODUL UTILIZAT:

```
*Procedura MEANS;
proc means data=hoteluri max min mean n nmiss sum;
  class City;
  var Price Rating ReviewsCount;
  title "Indicatori statistici pentru hoteluri, grupați pe oraș";
run;
```

REZULTATELE OBTINUTE:

Indicatori statistici pentru hoteluri, grupați pe orașe								
The MEANS Procedure								
City	N Obs	Variable	Maximum	Minimum	Mean	N	N Miss	Sum
Amsterdam City Center	12	Price Rating ReviewsCount	495.4200000 8.9000000 2974.00	134.7100000 7.0000000 500.0000000	259.0875000 8.2750000 1140.83	12	0	3109.05 99.3000000 13690.00
Amsterdam Noord	2	Price Rating ReviewsCount	167.9400000 8.4000000 778.0000000	86.7600000 8.1000000 500.0000000	127.3500000 8.2500000 639.0000000	2	0	254.7000000 16.5000000 1278.00
Bemelen	25	Price Rating ReviewsCount	311.6300000 10.0000000 1155.00	130.2200000 8.0000000 1.0000000	250.2712000 8.4080000 55.2800000	25	0	6256.78 210.2000000 1382.00
Boschstraatkwartier	2	Price Rating ReviewsCount	356.8700000 8.0000000 3772.00	277.5100000 7.5000000 500.0000000	317.1900000 7.7500000 2136.00	2	0	634.3800000 15.5000000 4272.00
Breda	20	Price Rating ReviewsCount	214.8200000 9.4000000 1232.00	43.8700000 6.5000000 36.0000000	103.9910000 8.2800000 570.1500000	20	0	2079.82 165.6000000 11403.00
Centrum	14	Price Rating ReviewsCount	302.2900000 9.1000000 8631.00	61.3000000 8.0000000 22.0000000	121.7714288 8.6571429 1145.57	14	0	1704.80 121.2000000 16038.00
City Centre	19	Price Rating ReviewsCount	171.5700000 9.4000000 3803.00	85.4900000 8.2000000 138.0000000	137.6152632 8.8157895 828.0000000	19	0	2614.69 167.5000000 15694.00

❖ *Interpretarea rezultatelor*

În zona Amsterdam City Center, analizând cele 12 hoteluri, observăm că prețurile variază între 134,71 și 495,42, cu o medie ridicată de aproximativ 259,09. Ratingul mediu al hotelurilor este de 8,28, iar numărul mediu de recenzii este foarte mare, peste 1100, ceea ce indică faptul că această zonă este una foarte populară și cu hoteluri bine cotate de turiști. În schimb, în Orasul Bemelen, unde sunt 25 de hoteluri, prețurile sunt similare ca nivel mediu, în jur de 250, dar numărul de recenzii este mult mai redus, cu o medie de doar 55, ceea ce sugerează că hotelurile sunt mai puțin frecventate, deși ratingul mediu este chiar puțin mai mare, în jur de 8,4. Pe de altă parte, în orașul Breda, cu 20 de hoteluri analizate, prețurile sunt semnificativ mai mici, media fiind de aproximativ 104, iar ratingul mediu este similar, în jur de 8,3. Numărul mediu de recenzii este moderat, ceea ce indică un echilibru între calitatea serviciilor și accesibilitatea prețurilor. Un caz particular este zona Delfshaven, unde există un singur hotel cu un preț accesibil de 74,73 și un rating foarte bun de 9,1, însă numărul de recenzii este mic, ceea ce limitează generalizarea concluziilor. În ansamblu, se poate observa o tendință ca hotelurile din zonele centrale sau turistice mari să aibă prețuri mai ridicate, un număr mare de recenzii și ratinguri bune, pe când zonele mai mici sau periferice prezintă prețuri mai accesibile, ratinguri similare sau ușor mai ridicate, dar o popularitate mai redusă reflectată printr-un număr mai mic de recenzii.

10. Generarea de grafice

❖ *Descrierea problemei*

Această procedură urmărește realizarea unui grafic vertical 3D care să prezinte distribuția numărului de hoteluri pe orașe. Scopul este vizualizarea clară și intuitivă a frecvenței apariției hotelurilor în fiecare oraș, pentru a putea identifica rapid zonele cu cea mai mare densitate de hoteluri.

❖ *Informații necesare pentru rezolvare*

Pentru această analiză, este necesar setul de date hoteluri, din care am utilizat variabila de interes City. Aceasta variabilă este folosită pentru a grupa hotelurile pe orașe și a calcula frecvența numărului de hoteluri din fiecare zonă.

❖ *Produs software / funcție / metodă de calcul folosită*

- Software: SAS

Functii si metode folosite:

- Funcția PROC GCHART
- Crearea unui grafic tip bară verticală 3D folosind opțiunea VBAR3D. Această funcție generează bare verticale pentru fiecare categorie (în cazul nostru, fiecare oraș), iar înălțimea barelor corespunde numărului de observații (hoteluri) pe fiecare oraș.

❖ *Rezolvarea cu ajutorul produsului software*

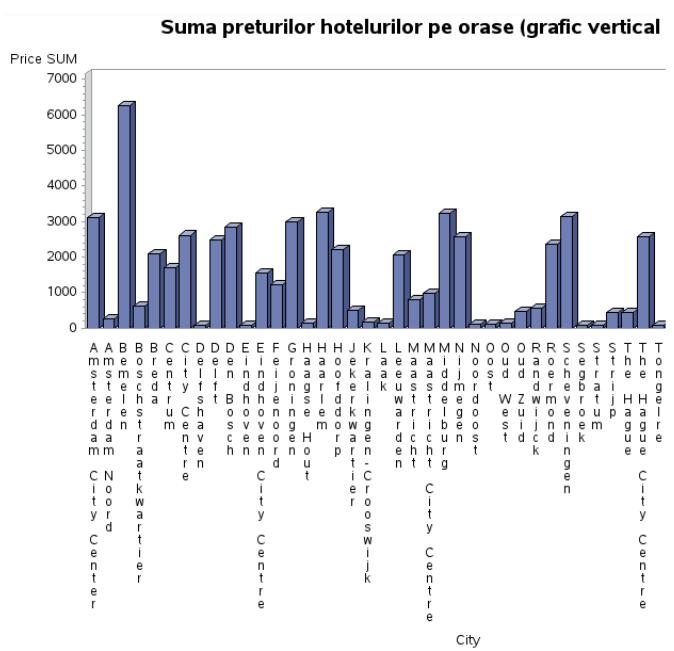
A fost utilizată instrucțiunea vbar3d City pentru a genera un grafic cu bare verticale 3D care evidențiază distribuția hotelurilor în funcție de oraș. Opțiunea type=freq a fost specificată pentru ca înălțimea fiecărei bare să reflecte frecvența, adică numărul de hoteluri din fiecare oraș. Prin sumvar=Price, graficul adaugă și o componentă cantitativă, însănd valorile variabilei Price pentru fiecare oraș, oferind astfel o dublă perspectivă: numerică și vizuală. Lățimea barelor a fost

ajustată la width=10 pentru claritate, iar opțiunea outline=black a fost folosită pentru a adăuga un contur negru barelor, sporind vizibilitatea și lizibilitatea graficului.

CODUL UTILIZAT:

```
*generarea de grafice;
proc gchart data=hoteluri;
vbar3d City /
discrete
type=freq
sumvar=Price
maxis=axis1
raxis=axis2
width=10
outline=black;
title "Suma prețurilor hotelurilor pe orașe (grafic vertical"
run;
quit;|
```

REZULTATELE OBTINUTE:



❖ Interpretarea rezultatelor

Graficul ilustrează distribuția prețurilor hotelurilor pe diferite orașe sau zone urbane, fiecare reprezentată printr-o bară verticală, iar pe axa verticală este indicată suma prețurilor (Price SUM). Se observă diferențe semnificative între orașe: unele zone, precum cea din partea stângă a graficului, au o valoare mult mai ridicată, ceea ce indică o sumă totală a prețurilor mai mare. Majoritatea zonelor au valori relativ mici, dar există câteva vârfuri notabile, ceea ce sugerează o concentrare a prețurilor hoteliere în anumite locații centrale.

11.Corelatii

11.1. Corelatia dintre Rating si Pret

❖ Descrierea problemei

Această procedură urmărește vizualizarea relației dintre scorul de satisfacție al clientilor (Rating) și prețul pe noapte al hotelurilor (Price). Scopul este de a identifica dacă există o corelație vizuală între cât plătesc clienții și cât de bine evaluează hotelurile, ceea ce poate indica dacă prețurile mai mari se corelează cu o calitate percepță superioară.

❖ Informații necesare pentru rezolvare

In aceasta analiza a fost utilizat setul de date hoteluri, utilizând urmatoarele variabile:

- **Rating** – scorul de satisfacție acordat hotelului de către clienți;
- **Price** – prețul pe noapte al hotelului

❖ Produs software / funcție / metodă de calcul folosită

- Software: SAS

Functii si metode folosite:

- Funcția PROC SGPlot

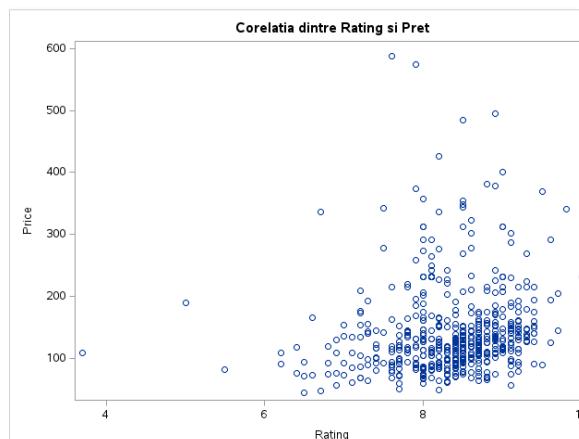
❖ Rezolvarea cu ajutorul produsului software

Instrucțiunea scatter x=Rating y=Price generează un grafic de dispersie în care valorile variabilei *Rating* sunt reprezentate pe axa orizontală (X), iar valorile variabilei *Price* pe axa verticală (Y), facilitând astfel vizualizarea relației dintre cele două variabile. De asemenea, comanda title este folosită pentru a adăuga un titlu sugestiv graficului, oferind context și claritate interpretării vizuale.

CODUL UTILIZAT:

```
*Corelatia dintre Rating si Preturi;
proc sgplot data=hoteluri;
    scatter x=Rating y=Price;
    title "Corelatia dintre Rating și Pret";
run;
```

REZULTATELE OBTINUTE:



❖ Interpretarea rezultatelor

Se observă că majoritatea hotelurilor au ratinguri între 7 și 9 și prețuri sub 200, însă există și hoteluri cu prețuri mult mai ridicate, unele depășind 500, chiar dacă ratingul lor nu este neapărat cel mai mare. Corelatia dintre cele 2 variabile nu pare a fi prea de puternic, punctele din grafic fiind destul de disperse.

11.2. Corelatia dintre numarul de recenzii si pret

❖ Descrierea problemei

Această procedură are ca scop vizualizarea relației dintre numărul de recenzii primite de hoteluri și prețul pe noapte. Scopul este de a observa dacă există o corelație între popularitatea unui hotel (măsurată prin numărul de recenzii) și nivelul prețului practicat.

❖ Informații necesare pentru rezolvare

In aceasta analiza a fost utilizat setul de date hoteluri, utilizand urmatoarele variabile:

- **ReviewsCount** – numărul total de recenzii primite de fiecare hotel;
- **Price** – prețul pe noapte al hotelului

❖ Produs software / funcție / metodă de calcul folosită

- **Software:** SAS

Functii si metode folosite:

- Funcția PROC GPLOT

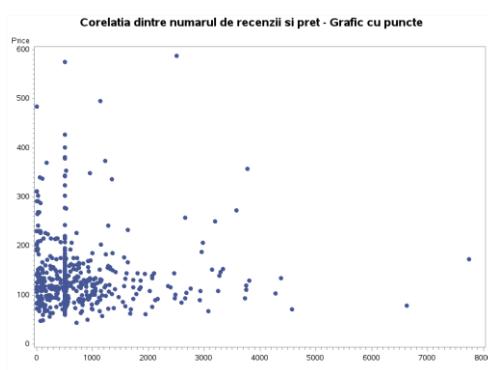
❖ Rezolvarea cu ajutorul produsului software

Instrucțiunea symbol value=dot; setează simbolul folosit pentru fiecare punct din grafic ca fiind un punct simplu (dot). Comanda plot Price * ReviewsCount; generează un grafic cu puncte în care valorile variabilei *Price* sunt reprezentate pe axa verticală, iar valorile variabilei *ReviewsCount* pe axa orizontală, permitând astfel analiza vizuală a relației dintre preț și numărul de recenzii. În plus, instrucțiunea title adaugă un titlu sugestiv graficului pentru o mai bună înțelegere a conținutului acestuia.

CODUL UTILIZAT:

```
*Corelatia dintre nr de recenzii si pret;
symbol value=dot;
title "Corelatia dintre numarul de recenzii si pret - Grafic cu puncte";
proc gplot data=hoteluri;
  plot Price * ReviewsCount;
run;
quit;
```

REZULTATELE OBTINUTE:



❖ *Interpretarea rezultatelor*

Graficul prezintă corelația dintre numărul de recenzii și prețul hotelurilor, fiecare punct reprezentând un hotel cu un anumit număr de recenzii (pe axa orizontală) și un anumit preț (pe axa verticală). Se observă că majoritatea hotelurilor au atât un număr relativ mic de recenzii (sub 1000), cât și prețuri moderate, în general sub 200. Există însă și hoteluri cu un număr foarte mare de recenzii, dar acestea nu au neapărat cele mai mari prețuri. De asemenea, hotelurile cu prețuri foarte mari (peste 300-400) tind să aibă mai puține recenzii. Nu se observă o corelație prea puternica între numărul de recenzii și preț: hotelurile cu multe recenzii nu sunt neapărat cele mai scumpe, iar cele mai scumpe nu au cele mai multe recenzii.

11.3. Procedura CORR

❖ *Descrierea problemei*

Această procedură urmărește să calculeze și să analizeze corelațiile statistice dintre variabilele Rating, Preț și Numărul de Recenzii ale hotelurilor. Scopul este de a identifica forța și direcția relațiilor liniare dintre aceste variabile, pentru a înțelege mai bine modul în care satisfacția clienților (Rating), prețul hotelurilor și popularitatea lor (numărul de recenzii) sunt legate între ele.

❖ *Informații necesare pentru rezolvare*

In aceasta analiza a fost utilizat setul de date hoteluri, utilizand urmatoarele variabile:

- **Rating** – scorul de satisfacție al clienților;
- **ReviewsCount** – numărul total de recenzii primite de fiecare hotel;
- **Price** – prețul pe noapte al hotelului

❖ *Produs software / funcție / metodă de calcul folosită*

- **Software:** SAS

Functii si metode folosite:

- Funcția PROC CORR

❖ *Rezolvarea cu ajutorul produsului software*

Procedura proc corr declanșează calculul corelațiilor între variabilele specificate. Prin utilizarea instrucțiunii var Rating Price ReviewsCount; se definesc variabilele între care se dorește calcularea coeficienților de corelație. De asemenea, titlul oferit raportului asigură o mai bună înțelegere și contextualizare a analizei realizate.

CODUL UTILIZAT:

```
*Procedura CORR;
title "Corelația dintre Rating, Preț și Numărul de Recenzii";
proc corr data=hoteluri;
  var Rating Price ReviewsCount;
run;
```

REZULTATELE OBTINUTE:

Corelația dintre Rating, Preț și Numărul de Recenzii

The CORR Procedure

3 Variables: Rating Price ReviewsCount

Simple Statistics						
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
Rating	525	8.32914	0.70536	4373	3.70000	10.00000
Price	525	141.99270	73.38523	74546	43.87000	587.83000
ReviewsCount	525	726.85333	872.73750	381073	1.00000	7748

Pearson Correlation Coefficients, N = 525 Prob > r under H0: Rho=0			
	Rating	Price	ReviewsCount
Rating	1.00000	0.14446 0.0009	-0.13999 0.0013
Price	0.14446 0.0009	1.00000	-0.05593 0.2007
ReviewsCount	-0.13999 0.0013	-0.05593 0.2007	1.00000

❖ Interpretarea rezultatelor

Rezultatele procedurii CORR indică faptul că există o corelație pozitivă slabă, dar semnificativă statistic, între Rating și Preț ($r = 0.144$, $p = 0.0009$), ceea ce sugerează că hotelurile cu ratinguri mai mari tind să aibă prețuri ușor mai ridicate. În schimb, corelația dintre Rating și Numărul de Recenzii este ușor negativă și semnificativă ($r = -0.140$, $p = 0.0013$), indicând că hotelurile cu mai multe recenzii tind să aibă un rating puțin mai scăzut. Corelația dintre Preț și Numărul de Recenzii nu este semnificativă ($r = -0.056$, $p = 0.2007$), ceea ce înseamnă că nu există o relație liniară clară între aceste două variabile. Aceste rezultate arată că, deși există unele legături între satisfacția clientilor, preț și popularitate, acestea sunt în general slabe și nu foarte puternice. Rating-ul hotelurilor are o medie de aproximativ 8.33 și o variație relativ mică (abatere standard de 0.70), ceea ce indică o satisfacție generală ridicată și destul de uniformă în rândul hotelurilor. Prețul mediu este în jur de 142 unități monetare, dar cu o dispersie mare (abatere standard 73.39), ceea ce reflectă o gamă largă de oferte de prețuri între hoteluri. Numărul mediu de recenzii este de aproximativ 726, însă cu o variație foarte mare (abatere standard 873), ceea ce arată că unele hoteluri sunt mult mai populare și primesc mult mai multe recenzii decât altele.

12. Regresie

12.1. Influenta rating-ului asupra pretului

❖ Descrierea problemei

Această procedură urmărește să analizeze influența variabilei Rating asupra prețului hotelurilor, prin construirea unui model de regresie liniară simplă. Scopul este de a determina dacă există o relație semnificativă între scorul de satisfacție al clientilor și prețul pe noapte, precum și de a evalua puterea acestei relații.

❖ Informații necesare pentru rezolvare

In aceasta analiza a fost utilizat setul de date hoteluri, utilizând urmatoarele variabile:

- **Rating** – scorul de satisfacție al clientilor;
- **Price** – prețul pe noapte al hotelului

❖ Produs software / funcție / metodă de calcul folosită

- **Software:** SAS

Functii si metode folosite:

- PROC PRINT
- Funcția PROC CORR
- PROC REG

❖ Rezolvarea cu ajutorul produsului software

PROC CORR calculează coeficientul de corelație Pearson între variabilele Rating și Price, pentru a confirma existența unei relații liniare între acestea. În continuare, PROC REG construiește un model de regresie liniară, în care Price este variabila dependentă, iar Rating este variabila explicativă. Graficul Price*Rating evidențiază dispersia punctelor și linia de regresie ajustată, oferind o reprezentare vizuală a relației dintre cele două variabile.

CODUL UTILIZAT:

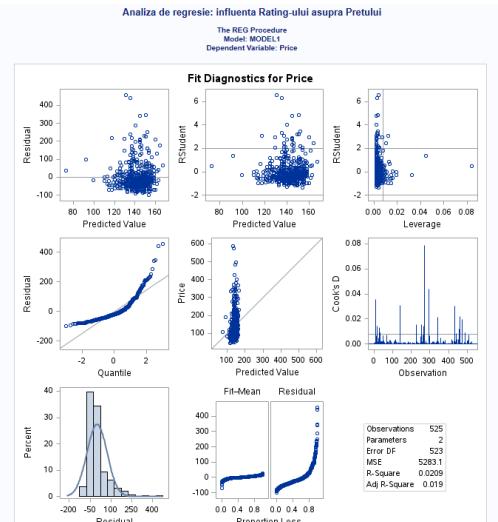
```
*Regresie
*Influenta Rating-ului asupra Pretului;
PROC PRINT DATA=hoteluri;
  TITLE "Vizualizare date hoteluri";
RUN;

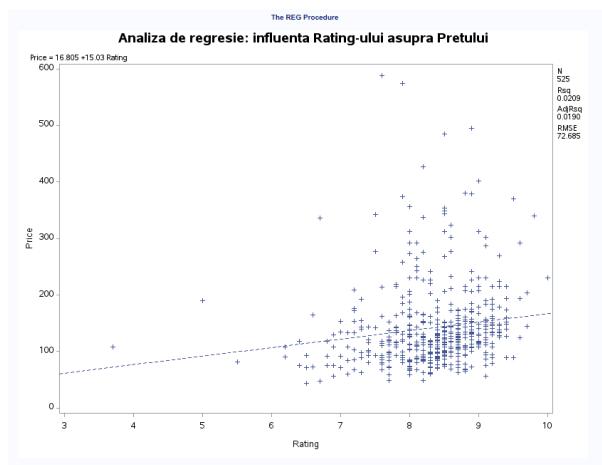
PROC CORR DATA=hoteluri;
  VAR Rating;
  WITH Price;
  TITLE "Corelatia dintre Rating si Pretul hotelurilor";
RUN;

PROC REG DATA=hoteluri;
  MODEL Price = Rating;
  PLOT Price*Rating;
  TITLE "Analiza de regresie: influenta Rating-ului asupra Pretului";
RUN;
```

REZULTATELE OBTINUTE:

Analiza de regresie: influenta Rating-ului asupra Pretului					
The REG Procedure					
Model: MODEL1					
Dependent Variable: Price					
Number of Observations Read 525					
Number of Observations Used 525					
Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	56894	56894	11.15	0.0009
Error	523	2783052	5283.08248		
Corrected Total	524	2821946			
Root MSE 72.68482 R-Square 0.0209					
Dependent Mean 141.99270 Adj R-Sq 0.0190					
Coeff Var 51.18912					
Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	16.80541	37.62887	0.45	0.6553
Rating	1	15.03003	4.50183	3.34	0.0009





❖ Interpretarea rezultatelor

Analiza de regresie efectuată evidențiază o influență pozitivă și semnificativă a rating-ului asupra prețului hotelurilor, cu un coeficient de 15,03 și un nivel de semnificație sub 1% ($p=0,0009$). Cu toate acestea, puterea explicativă a modelului este foarte redusă, R^2 fiind doar 2,09%, ceea ce indică faptul că majoritatea variațiilor nu poate fi explicată doar prin rating. Interceptul modelului este 16,81, iar coeficientul arată că o creștere cu o unitate a rating-ului corespunde unei majorări medii de aproximativ 15 unități monetare în preț. Analiza reziduurilor indică o dispersie mare și prezența unor valori extreme, sugerând că modelul nu surprinde complet variabilitatea prețurilor, iar distribuția reziduurilor nu este perfect normală. Graficul de dispersie confirmă o relație slabă pozitivă între rating și preț, dar cu o variabilitate considerabilă a prețurilor pentru același rating, iar linia de regresie aproape orizontală reflectă influența limitată a rating-ului asupra prețului.

12.2. Influenta numărului de recenzii asupra pretului

❖ Descrierea problemei

Această procedură analizează influența numărului de recenzii asupra prețului hotelurilor. Scopul este de a determina dacă numărul de recenzii poate explica variațiile prețului pe noapte și dacă există o relație semnificativă între aceste două variabile.

❖ Informații necesare pentru rezolvare

In aceasta analiza a fost utilizat setul de date hoteluri, utilizând urmatoarele variabile:

- ReviewsCount – numărul de recenzii primite de hotel;
- Price – prețul pe noapte al hotelului

❖ Produs software / funcție / metodă de calcul folosită

- Software: SAS

Functii si metode folosite:

- PROC REG;
- Plot – grafic de dispersie pentru vizualizarea relației dintre cele două variabile

❖ Rezolvarea cu ajutorul produsului software

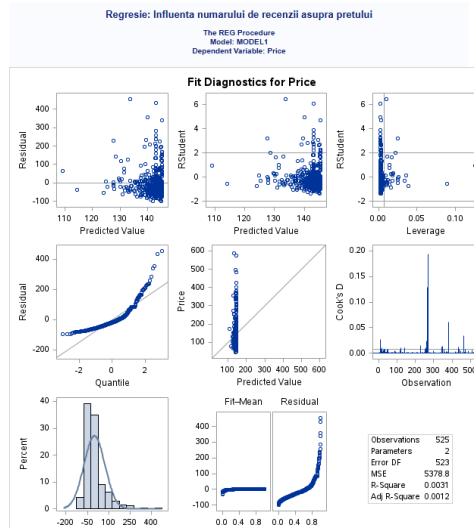
Procedura PROC REG construiește un model de regresie liniară în care variabila dependentă este Price (prețul), iar variabila explicativă este ReviewsCount (numărul de recenzii). Comanda model Price = ReviewsCount; specifică relația liniară dintre preț și numărul de recenzii. Instrucțiunea plot Price*ReviewsCount; generează graficul de dispersie care afișează relația dintre preț și numărul de recenzii, împreună cu linia de regresie estimată.

CODUL UTILIZAT:

```
*Influenta numarului de recenzii asupra pretului;
title "Regresie: Influenta numarului de recenzii asupra pretului";
proc reg data=hoteluri;
  model Price = ReviewsCount;
  plot Price ReviewsCount;
run;
quit;
```

REZULTATELE OBTINUTE:

Regresie: Influenta numarului de recenzii asupra pretului					
The REG Procedure Model: MODEL1 Dependent Variable: Price					
		Number of Observations Read	525		
		Number of Observations Used	525		
Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	8827.64693	8827.64693	1.64	0.2007
Error	523	2813118	5378.81080		
Corrected Total	524	2821946			
Root MSE 73.34038 R-Square 0.0031					
Dependent Mean 141.99270 Adj R-Sq 0.0012					
Coeff Var 51.65081					
Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	145.40638	4.16483	34.91	<.0001
ReviewsCount	1	-0.00470	0.00387	-1.28	0.2007



❖ Interpretarea rezultatelor

Rezultatele analizei de regresie pentru influența numărului de recenzii asupra prețului arată că modelul nu este semnificativ din punct de vedere statistic. Valoarea F este 1.64, cu un p-value de 0.2007, ceea ce indică faptul că nu există o relație semnificativă între numărul de recenzii și prețul hotelurilor la nivelul de semnificație de 5%. Coeficientul pentru ReviewsCount este negativ (-0.00470), dar nu este semnificativ statistic ($p=0.2007$), ceea ce înseamnă că influența numărului de recenzii asupra prețului este nesemnificativă și foarte slabă. Puterea explicativă a modelului este foarte redusă, cu un R-Square de doar 0.0031 (0.31%), ceea ce înseamnă că numărul de recenzii explică doar o foarte mică parte din variația prețului hotelurilor. Ajustarea R-Square confirmă această concluzie, având valoarea 0.0012. Interceptul are o valoare de 145.41, ceea ce reprezintă prețul mediu estimat atunci când numărul de recenzii este zero. Din graficele de dispersie rezultate în urma analizei de regresie reiese clar faptul că există o legătură

foarte slabă sau aproape inexistentă între numărul de recenzii și prețul hotelurilor, confirmând concluziile statistice ale modelului.

13.ANOVA

❖ *Descrierea problemei*

Această analiză își propune să investigheze dacă există diferențe semnificative între prețurile hotelurilor în funcție de orașul în care acestea sunt situate. Scopul este de a determina dacă locația (orașul) influențează prețul pe noapte al hotelurilor, comparând mediile prețurilor între mai multe grupuri (orașe).

❖ *Informații necesare pentru rezolvare*

In aceasta analiza a fost utilizat setul de date hoteluri, utilizând urmatoarele variabile:

- City – variabila categorică ce reprezintă orașul în care se află hotelul;
- Price – prețul pe noapte al hotelului

❖ *Produs software / funcție / metodă de calcul folosită*

- Software: SAS

Functii si metode folosite:

- PROC ANOVA, care testează dacă mediile prețurilor diferă semnificativ între grupurile definite de oraș;
Testul Tukey pentru compararea multiplă a mediilor, utilizat pentru a identifica care perechi de orașe au diferențe semnificative între prețuri.

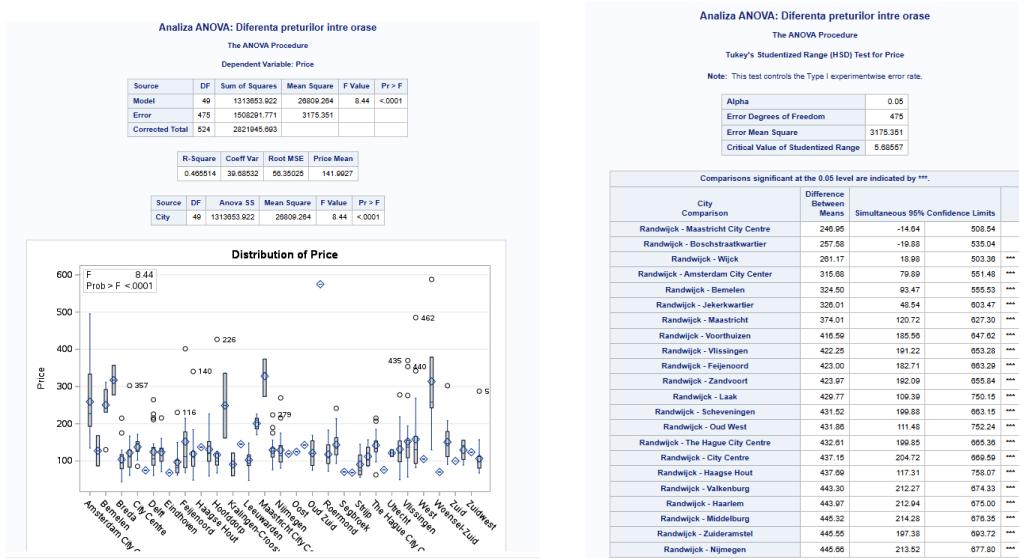
❖ *Rezolvarea cu ajutorul produsului software*

Instrucțiunea CLASS City grupează observațiile în funcție de oraș, iar comanda MODEL Price = City specifică un model în care prețul este explicitat de oraș. Opțiunea MEANS City / TUKEY realizează o comparație multiplă a mediilor între orașe folosind testul Tukey pentru diferențe semnificative. TITLE oferă un titlu descriptiv pentru rezultatele generate de procedură.

CODUL UTILIZAT:

```
*ANOVA;
PROC ANOVA DATA=hoteluri;
  CLASS City;
  MODEL Price = City;
  MEANS City / TUKEY;
  TITLE "Analiza ANOVA: Diferenta preturilor intre orase";
RUN;
```

REZULTATELE OBTINUTE:



❖ Interpretarea rezultatelor

Rezultatele analizei ANOVA indică existența unor diferențe semnificative statistic între prețurile hotelurilor din diferite orașe, având în vedere valoarea $p < 0.0001$ și un F Value de 8.44. Modelul explică aproximativ 46.55% din variația totală a prețurilor ($R^2 = 0.4655$), ceea ce arată o putere explicativă moderată a orașului asupra prețurilor. Testul de comparații multiple Tukey evidențiază că prețurile hotelurilor din orașul Randwijck diferă semnificativ față de cele din mai multe alte zone, cum ar fi Wijck, Amsterdam City Center, Bemelen, Jekerkwartier, Maastricht și Voorthuizen, unde diferențele medii sunt mari și intervalele de încredere nu includ zero.

PARTEA 3 – SAS ENTERPRISE GUIDE

1. Importul unui fisier Excel

❖ Descrierea problemei

Primul pas în cadrul proiectului a avut ca scop preluarea și integrarea într-un mediu de analiză a datelor referitoare la hoteluri, disponibile într-un fișier Excel. Conversia acestora într-un set de date SAS a fost necesară pentru a permite aplicarea ulterioară a tehniciilor de analiză statistică și generarea de rapoarte relevante în SAS Enterprise Guide.

❖ Informații necesare pentru rezolvare

- Fișierul sursă: HotelDataset_final.xlsx;
- Structura datelor:- Variabile numerice: ID, Price, ReviewsCount, Rating;
 - Variabile de tip text (string): Name, Type, City, State;

- Număr total de observații: 525.

❖ Produs software / Funcție / Metodă de calcul folosită

- Produs: SAS Enterprise Guide 8.3;

- Funcționalitate: File → Import Data;

- Task suplimentar: Describe → Characterize Data pentru analizarea variabilelor din setul importat.

❖ Rezolvarea cu ajutorul produsului software

Pentru a începe analiza datelor, s-a creat un nou proiect în SAS Enterprise Guide, iar din bara de opțiuni a fost selectată comanda File → Import Data. A fost ales fișierul HotelDataset_final.xlsx de pe sistemul local, iar în primul pas al expertului de import s-a specificat sursa datelor și setul de ieșire (output data set), reprezentând locația și denumirea datasetului SAS generat. În pasul al doilea, s-a bifat opțiunea "First row of range contains field names", astfel încât anteturile coloanelor din Excel să fie recunoscute ca nume de variabile. La pasul patru, a fost selectată opțiunea "Embed the data within the generated SAS code", pentru a include datele în codul SAS rezultat. După finalizarea importului, s-a obținut un set de date cu 332 de observații și 8 variabile, dintre care unele numerice (ID, Price, ReviewsCount, Rating) și altele de tip string (Name, Type, City, State). Pentru a realiza o primă analiză descriptivă a datasetului, s-a utilizat task-ul Tasks → Describe → Characterize Data, bifând opțiunile Summary, Graphs și SAS Data Sets, cu limitarea afișării la maximum 30 dintre cele mai frecvente valori distincte pentru variabilele categorice.

The screenshot shows the SAS Enterprise Guide interface. On the left, a window titled 'Import Data from HotelDataset_SAS.xlsx' displays the 'Define Field Attributes' step, where columns from the Excel file are mapped to SAS variables. On the right, a process flow diagram illustrates the data pipeline: 'HotelDataset_SAS.xlsx' → 'Import Data (HotelDataset_SAS.xls)' → 'Data Imported from Hotel' → 'Characterize Data' → 'Frequency Counts for WORK.HO...'. From 'Characterize Data', arrows point to 'Univariate Statistics for WORK.HO...' and 'HTML - Characterize Data'.

The screenshot shows the results of the 'Characterize Data' task. It displays a table with 10 rows of hotel information, including ID, Name, Type, Price, ReviewsCount, Rating, City, and State. The table highlights the 'ID' column.

	ID	Name	Type	Price	ReviewsCount	Rating	City	State
1	0	BUNK Hotel Amsterdam	Bunk Pod for 2	86.76	778	8.4	Amsterdam Noord	Amsterdam
2	1	YOTEL Amsterdam	Premium Double Room	167.94	500	8.1	Amsterdam Noord	Amsterdam
3	2	Multatului Hotel	Double Room	143.69	1605	7.4	Amsterdam City Center	Amsterdam
4	3	nhow Amsterdam Rai	nhow Double or Twin Room with View	141.39	500	9.0	Zuideramstel	Amsterdam
5	4	Motel One Amsterdam	Double Room	104.18	500	8.8	Zuideramstel	Amsterdam
6	5	INNSIDE by Meliá Amsterdam	The Innside Guestroom	155.35	1264	8.4	Zuideramstel	Amsterdam
7	6	Eden Hotel Amsterdam	Small Double Room	220.66	500	8.3	Amsterdam City Center	Amsterdam
8	7	citizenM Amsterdam South	King Room	156.27	500	8.8	Zuideramstel	Amsterdam
9	8	The Alfred Hotel	Standard Double Room	139.21	2069	7.3	Oud Zuid	Amsterdam
10	9	Andaz Amsterdam Prinsengracht - a concept by Hyatt	Observatory King	495.42	1140	8.9	Amsterdam City Center	Amsterdam

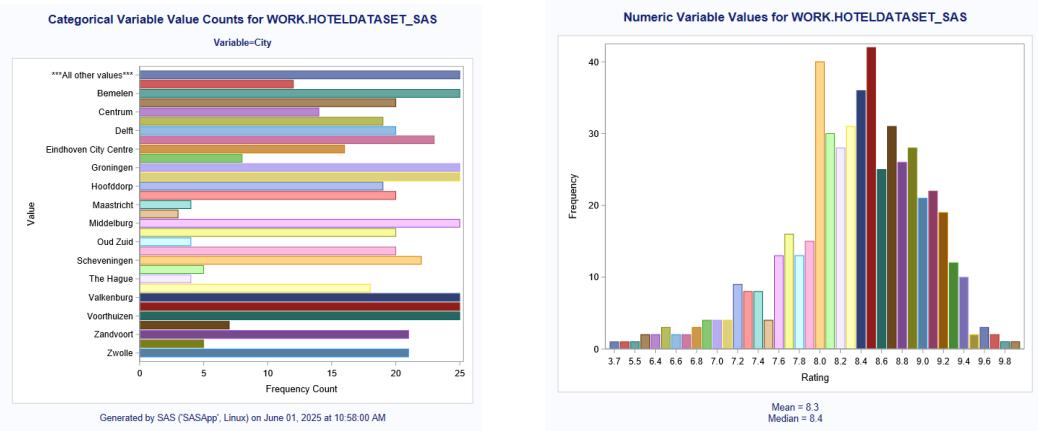
Summary of Categorical Variables for WORK.HOTELDATASET_SAS
Limited to the 30 Most Frequent Distinct Values per Variable

Variable	Label	Value	Frequency Count	Percent of Total Frequency
City	Bemelen	25	4.7619	
	Groningen	25	4.7619	
	Haarlem	25	4.7619	
	Middelburg	25	4.7619	
	Valkenburg	25	4.7619	
	Vlissingen	25	4.7619	
	Voorhuzen	25	4.7619	
	Den Bosch	23	4.3810	
	Scheveningen	22	4.1905	
	Zandvoort	21	4.0000	
	Zwolle	21	4.0000	
	Breda	20	3.8095	
	Delft	20	3.8095	
	Leeuwarden	20	3.8095	
	Nijmegen	20	3.8095	
	Roermond	20	3.8095	
	City Centre	19	3.6190	
	Hoofddorp	19	3.6190	
	The Hague City Centre	18	3.4286	
	Eindhoven City Centre	16	3.0476	
	Centrum	14	2.6667	
	Amsterdam City Center	12	2.2857	
	Feijenoord	8	1.5238	
	Wijk	7	1.3333	
	Strijp	5	0.9524	
	Zuidasrmstel	5	0.9524	
	Maastricht	4	0.7619	
	Oud Zuid	4	0.7619	
	The Hague	4	0.7619	
	Maastricht City Centre	3	0.5714	
	All other values	25	4.7619	

Summary of Numeric Variables for WORK.HOTELDATASET_SAS

Variable	Label	N	NMiss	Total	Min	Mean	Median	Max	StdMean
ID		525	0	137550.00	0.00	262.000	262.0	524.00	6.6207
Price		525	0	74546.17	43.87	141.993	124.7	587.83	3.2028
Rating		525	0	4372.80	3.70	8.329	8.4	10.00	0.0308
ReviewsCount		525	0	381073.00	1.00	725.853	500.0	7748.00	38.0894

Generated by SAS ('SASApp', Linux) on June 01, 2025 at 10:58:00 AM



❖ Interpretarea rezultatelor

În cazul variabilelor categorice, se observă o concentrare relativ uniformă a hotelurilor în anumite orașe precum Bemelen, Groningen și Haarlem, fiecare reprezentând aproximativ 4,76% din total, iar statul Limburg se remarcă drept regiunea cu cea mai mare frecvență de hoteluri (13,33). Variabila ID este o cheie unică generată automat, variind de la 0 la 524, fără valori lipsă, ceea ce indică o numerotare completă și corectă a celor 525 de înregistrări. În ceea ce privește Price, prețul cazării variază între 43,87 și 587,83, cu o medie de 141,99 și o mediană de 124,7, ceea ce sugerează o distribuție ușor asimetrică, existând hoteluri cu prețuri semnificativ mai mari decât media. Aceasta poate reflecta diferențe de confort, localizare sau reputație. Variabila Rating, cuprinsă între 3,7 și 10, are o medie de 8,33 și o mediană de 8,4, ceea ce indică faptul că majoritatea hotelurilor au evaluări bune spre foarte bune, un semnal pozitiv pentru companie. ReviewsCount variază între 1 și 7748, cu o medie de 725,85 și o mediană de 500, demonstrând o dispersie mare și sugerând că unele hoteluri sunt semnificativ mai populare sau mai bine promovate decât altele.

2. Interogări

2.1. Selectarea coloanelor si filtrarea campurilor

❖ *Descrierea problemei*

Scopul acestei interogări este de a identifica distribuția hotelurilor din Amsterdam în afara zonei centrale Amsterdam City Center, pentru a evidenția concentrarea unităților de cazare în cartierele adiacente. Această analiză este utilă pentru o mai bună înțelegere a potențialului turistic și a gradului de dezvoltare în diferite zone urbane ale orașului..

❖ *Informații necesare pentru rezolvare*

Pentru a răspunde la această întrebare, sunt necesare următoarele câmpuri din setul de date:

- City – zona sau cartierul din cadrul orașului Amsterdam;
- State – orașul propriu-zis, utilizat pentru filtrarea doar a înregistrărilor relevante;
- ID – identificator unic pentru fiecare hotel, necesar pentru aplicarea funcției de numărare.

❖ *Produs software / Funcție / Metodă de calcul folosită*

- Produs: SAS Enterprise Guide 8.3;
- Funcția de agregare COUNT aplicată pe coloana ID;
- Clauza WHERE pentru filtrarea datelor (State = 'Amsterdam' și City \neq 'Amsterdam City Center');
- Selecția coloanelor City, State și rezultatul agregării (COUNT_of_ID).

❖ *Rezolvarea cu ajutorul produsului software*

În cadrul SAS EG, am pornit de la tabela HOTELDATASET_SAS și am construit un query în care am selectat coloanele ID,City și State. Pentru a calcula numărul de hoteluri din fiecare zonă, am aplicat funcția Count(ID) și s-a redenumit rezultatul în COUNT_of_ID. A fost configurată o clauză de filtrare care a inclus doar observațiile din orașul Amsterdam, excluzând zona „Amsterdam City Center”. Apoi am grupat rezultatele după coloana City, ceea ce a permis identificarea numărului de hoteluri pentru fiecare zonă din Amsterdam, exceptând centrul orașului.

	City	State	COUNT_of_ID
1	Amsterdam Noord	Amsterdam	2
2	Oost	Amsterdam	1
3	Oud West	Amsterdam	1
4	Oud Zuid	Amsterdam	4
5	Zuideramstel	Amsterdam	5

❖ Interpretarea rezultatelor

Rezultatele au evidențiat faptul că zonele cu cel mai mare număr de hoteluri, în afara centrului, sunt Zuideramstel (5 hoteluri) și Oud Zuid (4 hoteluri). Alte zone, precum Amsterdam Noord, Oost și Oud West, înregistrează un număr mai redus de hoteluri, între 1 și 2. Aceste informații indică o distribuție inegală a capacitatilor de cazare în cartierele Amsterdamului.

2.2. Crearea unei coloane calculate

❖ Descrierea problemei

Pentru a obține o măsură mai relevantă a calității unui hotel, am realizat ajustarea scorului de rating astfel încât să reflecte nu doar nota acordată, ci și încrederea conferită de numărul de recenzii primite. Scopul a fost evitarea supraevaluării hotelurilor cu scoruri mari, dar cu puține recenzii, prin introducerea unui indicator compozit – scorul ajustat.

❖ Informații necesare pentru rezolvare

Pentru acest calcul au fost necesare două câmpuri existente în setul de date:

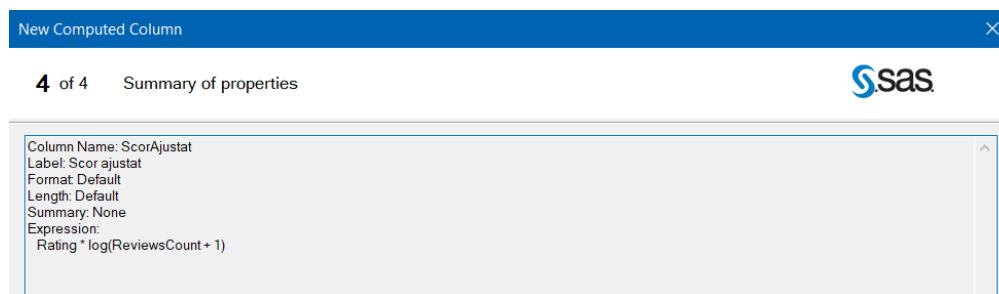
- Rating – scorul acordat hotelului de către clienți (variabilă numerică între 3.7 și 10);
- ReviewsCount – numărul total de recenzii primite de hotel.

❖ Produs software / Funcție / Metodă de calcul folosită

- Produs: SAS Enterprise Guide 8.3;
- Funcționalitatea **New Computed Column**;
- Funcția matematică log() (logaritm natural);
- Operatorul de multiplicare *;
- Formula utilizată: ScorAjustat = Rating * log(ReviewsCount + 1).

❖ Rezolvarea cu ajutorul produsului software

În SAS Enterprise Guide, am accesat funcția New Computed Column, unde am definit o nouă variabilă numită ScorAjustat. Aceasta a fost calculată pentru fiecare observație din setul de date HOTELDATASET_SAS, folosind expresia matematică menționată. Noua coloană a fost adăugată în cadrul aceluiași tabel, fără a fi necesară vreo filtrare suplimentară a datelor.



ID	Name	ReviewsCount	Rating	ScorAjustat
1	BUNK Hotel Amsterdam	778	8.4	55.927292785
2	YOTEL Amsterdam	500	8.1	50.354509419
3	Multatuli Hotel	1605	7.4	54.623114019
4	nhow Amsterdam Rai	500	9.0	55.94945491
5	Motel One Amsterdam	500	8.8	54.70613369
6	INNSiDE by Meliá Amsterdam	1264	8.4	59.99975017
7	Eden Hotel Amsterdam	500	8.3	51.597830639
8	citizenM Amsterdam South	500	8.8	54.70613369
9	The Alfred Hotel	2069	7.3	55.73771837
10	Andaz Amsterdam Prinsengracht - a concept by Hyatt	1140	8.9	62.652977114

❖ Interpretarea rezultatelor

Coloana ScorAjustat permite o ierarhizare mai echitabilă a hotelurilor, deoarece ia în considerare atât scorul acordat, cât și volumul de recenzii – element esențial pentru validitatea evaluării. Valorile acestei noi variabile au variat între 3.46 și 77.47, indicând o variație semnificativă între hotelurile cu scoruri susținute de un număr mare de recenzii și cele cu scoruri potențial înselătoare, bazate pe puțini clienți. Această ajustare contribuie la o analiză mai robustă a performanței hotelurilor.

2.3. Crearea unei coloane recodificate

❖ Descrierea problemei

Pentru o interpretare mai clară și mai accesibilă a scorurilor ajustate ale hotelurilor, s-a urmărit gruparea acestora în categorii semnificative (Scor Scăzut, Mediu și Ridicat). Această recodificare permite segmentarea mai ușoară a hotelurilor pe baza performanței percepute și facilitează analiza comparativă și vizuală a distribuției scorurilor.

❖ Informații necesare pentru rezolvare

Pentru a rezolva aceasta problema, am utilizat tabela „Scor Ajustat” :

- am determinat valorile minime și maxime ale acestei coloane (obținute cu opțiunea Summary Statistics);

Summary Statistics Results	
The MEANS Procedure	
Analysis Variable : ScorAjustat Scor ajustat	
Minimum	Maximum
3.4657359	77.4370229

- Criteriile de grupare în categorii:

- 3 – <30.0 → Scor Scăzut;
- 30.01 – <60.0 → Scor Mediu;
- ≥60.01 → Scor Ridicat.

❖ *Produs software / Funcție / Metodă de calcul folosită*

- Produs: SAS Enterprise Guide 8.3;
- Browse → Describe → Summary Statistics – pentru obținerea valorilor MIN și MAX ale coloanei ScorAjustat;
- New Computed Column :
 - Funcționalitate: Replacement;
 - Funcții utilizate:- Comparatori logici (<, >=);
 - Atribuirি condiționale în intervale de valori;
- Tasks → Describe → One-Way Frequencies;
 - Setări folosite:- Analysis variable: Categorie_scor;
 - Selectate opțiuni: Frequency, Percent, Cumulative Frequency, Cumulative Percent;
- Graphs:- Vertical Bar Chart (Categorie_scor vs. Frequency);
 - Cumulative Distribution chart;
 - Deviations plot (Relative Deviation vs. Categorie_scor);
- Chi-Square Test for Equal Proportions.

❖ *Rezolvarea cu ajutorul produsului software*

După determinarea valorilor minime și maxime ale scorului ajustat folosind Summary Statistics, a fost creată o nouă coloană cu numele Categorie_scor în cadrul tabelului Scor_ajustat, utilizând New Computed Column. În secțiunea Replacement, am definit trei intervale pe baza valorilor scorului ajustat: scoruri mai mici de 30, dar mai mari decat 3 au fost clasificate ca „Scor Scăzut”, scorurile între 30.01 și 60 ca „Scor Mediu”, iar cele mai mari sau egale cu 60.01 ca „Scor Ridicat”.

Ulterior, s-a utilizat task-ul One-Way Frequencies pentru a calcula frecvențele aferente fiecărei categorii. La setările task-ului, s-a selectat variabila Categorie_scor ca Analysis Variable și s-au bifat opțiunile pentru Frequency, Percent, Cumulative Frequency și Cumulative Percent. S-a obținut o distribuție clară:

- Scor Mediu: 389 observații
- Scor Ridicat: 75 observații
- Scor Scăzut: 61 observații

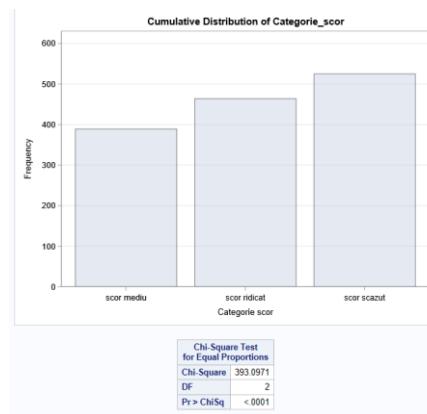
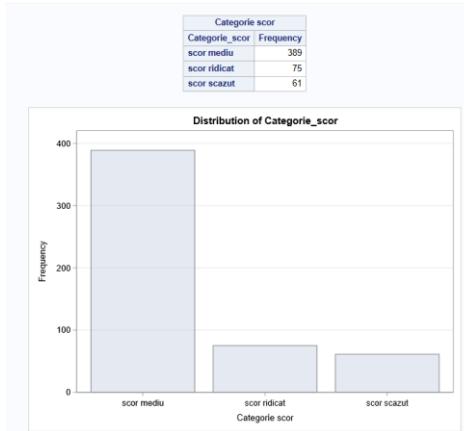
Au fost generate și două grafice: unul cu bare verticale pentru frecvențele simple, cu Categorie_scor pe axa X și Frequency pe axa Y (0–400), și unul cu distribuția cumulativă. În plus, s-a rulat testul Chi-Square for Equal Proportions, pentru a verifica dacă cele trei categorii apar cu frecvențe egale.

5 of 5 Summary of properties

```

Column Name: Categorie_scor
Label: Categorie scor
Type: Character
Format: Default
Length: Default
Summary: None
Expression:
CASE
WHEN t1.ScorAjustat >= 60.01 THEN 'scor ridicat'
WHEN t1.ScorAjustat >= 3 AND t1.ScorAjustat <= 30 THEN 'scor scăzut'
WHEN t1.ScorAjustat >= 30.01 AND t1.ScorAjustat <= 60 THEN 'scor mediu'
END
  
```

ID	Name	ScorAjustat	Categorie_scor
1	0 BUNK Hotel Amsterdam	55.927292785	scor mediu
2	1 YOTEL Amsterdam	50.354509419	scor mediu
3	2 Multatuli Hotel	54.623114019	scor mediu
4	3 nhow Amsterdam Rai	55.94945491	scor mediu
5	4 Motel One Amsterdam	54.70613369	scor mediu
6	5 INNSIDE by Meliá Amsterdam	59.99975017	scor mediu
7	6 Eden Hotel Amsterdam	51.597830639	scor mediu
8	7 citizenM Amsterdam South	54.70613369	scor mediu
9	8 The Alfred Hotel	55.73771837	scor mediu
10	9 Andaz Amsterdam Prinsengracht - a concept by Hyatt	62.652977114	scor ridicat



❖ Interpretarea rezultatelor

Rezultatele evidențiază o distribuție inegală a scorurilor ajustate: majoritatea hotelurilor (389 din 525) sunt încadrate în categoria „Scor Mediu”, în timp ce doar 75 au un scor ridicat și 61 un scor scăzut. Testul Chi-Square a confirmat că aceste diferențe sunt statistic semnificative (Chi-Square = 393.0971, df = 2, p < 0.001), indicând că scorurile nu sunt distribuite uniform între categorii. În graficul de deviații, doar scorul mediu are o abatere relativă pozitivă (>1), ceea ce confirmă că această categorie este supra-reprezentată, în timp ce scorurile scăzute și ridicate sunt sub-reprezentate (deviații < -0.5). Această clasificare permite o înțelegere rapidă a calității generale a hotelurilor și poate fi utilă în analize de segmentare sau în luarea deciziilor de business.

2.4. Jonctiuni

❖ Descrierea problemei

Pentru a facilita segmentarea ofertelor hoteliere în funcție de scorul ajustat obținut anterior, a fost necesară alăturarea (join-ul) între datele inițiale și categoriile de scor, precum și crearea unei coloane recodificate care să exprime pachetele hoteliere în termeni de ofertă comercială (ex: „Premium Package”, „Standard Deal”, „Budget Choice”). Această etapă are ca scop conversia informațiilor numerice în etichete semnificative.

❖ Informații necesare pentru rezolvare

- Tabela inițială: HOTELDATASET_SAS;
- Tabela cu categorii scor: conține coloana Categorie_scor;

- Reguli de recodificare:

- Scor Ridicat → Premium Package;
- Scor Mediu → Standard Deal;
- Scor Scăzut → Budget Choice.

❖ *Produs software / Funcție / Metodă de calcul folosită*

- Produs: SAS Enterprise Guide 8.3;

- Join (SQL Join) între două tabele;

- Edit Computed Column:

- Tip expresie: CASE (condițională);

- Setări specifice:- Setat ca tip Character;

- Activată opțiunea pentru missing value handling;

- Reguli de recodificare cu expresii CASE:

- WHEN Categorie_scor = "Scor Ridicat" THEN "Premium Package";
- WHEN Categorie_scor = "Scor Mediu" THEN "Standard Deal";
- WHEN Categorie_scor = "Scor Scăzut" THEN "Budget Choice";

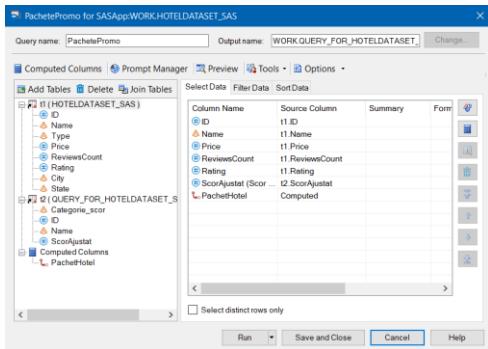
❖ *Rezolvarea cu ajutorul produsului software*

Am realizat o joncțiune între tabela inițială HOTELDATASET_SAS și tabela derivată care conține Categorie_sco. În urma acestui join, am creat un nou Query ce conține următoarele coloane: ID, Name, Price, ReviewsCount, Rating, ScorAjustat și Categorie_scor.

Apoi, prin opțiunea Edit Computed Column, am adăugat o nouă coloană calculată de tip Character, denumită PachetHotel, care recodifică valorile din Categorie_scor în trei etichete semnificative:

- „Scor Ridicat” → „Premium Package”
- „Scor Mediu” → „Standard Deal”
- „Scor Scăzut” → „Budget Choice”

Pentru definirea expresiei am folosit funcția CASE, iar în opțiunile avansate s-a bifat și gestionarea valorilor lipsă, pentru a preveni erori la rulare. Astfel, s-a obținut un query final cu informații atât cantitative (scor, preț, rating), cât și calitative (tipul pachetului).



ID	Name	Price	ReviewsCount	Rating	ScorAjustat	PachetHotel
0	BUNK Hotel Amsterdam	86.76	778	8.4	55.927292785	Standard Deal
1	YOTEL Amsterdam	167.94	500	8.1	50.354509419	Standard Deal
2	Multatul Hotel	143.69	1605	7.4	54.623114019	Standard Deal
3	nhow Amsterdam Rai	141.39	500	9.0	55.94945491	Standard Deal
4	Motel One Amsterdam	104.18	500	8.8	54.70613369	Standard Deal
5	INNSIDE by MeliÁ Hotel Amsterdam	155.35	1264	8.4	59.99975017	Standard Deal
6	Eden Hotel Amsterdam	220.66	500	8.3	51.597830639	Standard Deal
7	citizenM Amsterdam South	156.27	500	8.8	54.70613369	Standard Deal
8	The Alfred Hotel	139.21	2069	7.3	55.73771837	Standard Deal
9	Andaz Amsterdam Prinsengracht - a concept by Hyatt	495.42	1140	8.9	62.652977114	Premium Package
10	Hyatt Regency Amsterdam	343.51	500	8.5	52.841151859	Standard Deal

❖ Interpretarea rezultatelor

Rezultatul jonctiunii și al recodificării aduce o valoare suplimentară în analiza datelor: fiecare hotel este acum asociat cu un pachet de ofertă corespunzător performanței sale. Etichetele „Premium Package”, „Standard Deal” și „Budget Choice” oferă un mod clar de a comunica nivelul serviciilor și satisfacției clienților, bazat pe datele reale de recenzii și scoruri ajustate.

2.5. Interogari cu parametri

❖ Descrierea problemei

Interogările cu parametri vizează problema filtrării și selecției dinamice a datelor în funcție de criterii flexibile, stabilite de utilizator în momentul rulării interogării. Scopul acestor interogări în analiza realizată este de a permite extragerea rapidă și precisă a unui subset relevant de date, cum ar fi hotelurile care indeplinesc anumite criterii. Astfel, utilizatorul poate ajusta pragurile de selecție fără a modifica manual codul sau structura interogării, ceea ce facilitează explorarea și raportarea datelor în funcție de nevoi variabile și analize mai eficiente.

❖ Informații necesare pentru rezolvare

Pentru rezolvarea acestui task este necesara tabela inițială „PachetePromotionale” care conține coloanele: ID, Name, Price, ReviewsCount, Rating, ScorAjustat, PachetHotel.

❖ Produs software / funcție / metodă de calcul folosită

- **Software:** SAS Enterprise Guide

Functii si metode folosite:

- Summary Statistics Wizard - folosit pentru agregarea datelor din tabelul „PachetePromotionale”, calculând suma (Price_Sum) și media (Price_Mean) prețurilor pe grupuri după variabila „PachetHotel”;
- Parameter Manager – utilizat pentru crearea parametrului numeric „PriceLimit”, care permite filtrarea dinamică a rezultatelor în funcție de o valoare introdusă de utilizator;
- Query Builder (filtrare cu parametru) – metoda aplicată pentru setarea unui filtru pe coloana agregată (Price_Sum), folosind parametrul „PriceLimit” pentru a afișa doar înregistrările care depășesc pragul definit;
- Run Query - funcția de rulare a interogării;
- Join între tabele - pentru asocierea detaliilor hotelurilor cu pachetele promționale, folosind coloana „PachetHotel” ca legătură între tabela inițială și cea agregată;

❖ Rezolvarea cu ajutorul produsului software

Pentru rezolvarea task-ului, mai întâi s-a agregat tabelul „PachetePromotionale” folosind Summary Statistics Wizard, calculând suma totală și media prețurilor pe fiecare grupă „PachetHotel”. Apoi, s-a creat un parametru numeric „PriceLimit” în Parameter Manager, care permite utilizatorului să introducă un prag minim pentru suma prețurilor. Ulterior, în Query Builder s-a aplicat un filtru dinamic pe coloana cu suma prețurilor, astfel încât să fie afișate doar pachetele cu totalul prețurilor mai mare decât valoarea parametrului. În final, interogarea a fost rulată, iar la rulare s-a introdus pragul dorit (500), rezultatul afișând doar pachetele promotionale care respectă această condiție. Optional, am realizat o asociere între tabela agregată și cea inițială pentru a vizualiza detalii suplimentare despre hoteluri.

Rezultat Summary Statistics Wizard:

The screenshot shows the SAS Summary Statistics Wizard interface. On the left, under "Available variables", there is a list of columns: ID, Name, Price, ReviewsCount, Rating, ScoreAustat, and PachetHotel. In the center, under "Summary statistics of (Analysis variable)", the "Price" column is selected. Below it, under "For each value of (Classification variable)", the "PachetHotel" column is selected. To the right, a preview table displays the results:

Pachet promotional	Price_Sum	Price_Mean
Budget Choice	11778.92	193.10
Premium Package	10493.28	139.91
Standard Deal	52273.97	134.38

Rezultat JOIN:

ID	Name	PachetHotel	Price_Sum	Price_Mean
1	BUNK Hotel Amsterdam	Standard Deal	52273.97	134.38
2	YOTEL Amsterdam	Standard Deal	52273.97	134.38
3	Multatului Hotel	Standard Deal	52273.97	134.38
4	nhow Amsterdam Rai	Standard Deal	52273.97	134.38
5	Motel One Amsterdam	Standard Deal	52273.97	134.38
6	INNSiDE by Meliá Amsterdam	Standard Deal	52273.97	134.38
7	Eden Hotel Amsterdam	Standard Deal	52273.97	134.38
8	citizenM Amsterdam South	Standard Deal	52273.97	134.38
9	The Alfred Hotel	Standard Deal	52273.97	134.38
10	Andaz Amsterdam Prinsengracht - a concept by Hyatt	Premium Package	10493.28	139.91

3. Prelucrarea datelor

3.1. Crearea unui raport lista

❖ Descrierea problemei

Problema vizată de crearea unui raport listă constă în organizarea și prezentarea clară a informațiilor despre hoteluri, astfel încât să fie ușor de analizat și comparat. Prin gruparea hotelurilor pe orașe și afișarea prețurilor aferente fiecărui hotel, raportul oferă o imagine structurată asupra distribuției ofertelor de cazare și a nivelurilor de preț în funcție de locație.

❖ *Informații necesare pentru rezolvare*

Pentru a crea un raport lista cu hotelurile grupate pe orașe și afișarea prețurilor aferente, a fost necesar tabelul SAS utilizat, care contine coloanele ID, Name si Price.

❖ *Produs software / funcție / metodă de calcul folosită*

- **Software:** SAS Enterprise Guide

Functii si metode folosite:

- Task -> List Data – pentru a genera un raport de tip listă cu înregistrările selectate, permitând afișarea variabilelor relevante și gruparea acestora;

❖ *Rezolvarea cu ajutorul produsului software*

Am generat un raport lista pentru a afisa hotelurile grupate în funcție de oraș, împreună cu prețurile aferente fiecărui hotel, pornind de la tabela care contine datele necesare. Apoi, cu ajutorul opțiunii List Data am setat inclus campurile Name si Price in raport pentru a vedea denumirea și prețul fiecărui hotel și am grupat după variabila City, astfel incat hotelurile să fie afișate separat pentru fiecare oraș. Astfel, s-a generat un raport care oferă o privire de ansamblu asupra hotelurilor din diferite orase, utila pentru analizele comparative între orașe.

List Data for WORK.HOTELDATASET_SAS			
City-Amsterdam City Center			
Obs	ID	Name	Price
1	2	Multatuli Hotel	143.69
2	6	Eden Hotel Amsterdam	220.66
3	9	Andaz Amsterdam Prinsengracht - a concept by Hyatt	495.42
4	10	Hyatt Regency Amsterdam	343.51
5	11	NH Collection Amsterdam Flower Market	323.30
6	12	ibis Styles Amsterdam Central Station	195.78
7	14	Rho Hotel	206.56
8	16	Hotel V Nesplein	241.09
9	17	Kimpton De Witt Amsterdam, an IHG Hotel	380.79
10	21	The White Tulip Hostel	134.71
11	22	Swissotel Amsterdam	233.15
12	23	Holiday Inn Express Amsterdam - City Hall, an IHG Hotel	190.39
City-Amsterdam Noord			
Obs	ID	Name	Price
13	0	BUNK Hotel Amsterdam	86.76
14	1	YOTEL Amsterdam	167.94
City-Bemelen			
Obs	ID	Name	Price
15	25	Holiday Home Green Resort Mooi Bemelen-19	230.80
16	26	Holiday Home Green Resort Mooi Bemelen-2	291.87
17	27	Holiday Home Green Resort Mooi Bemelen-12	230.80
18	28	Holiday Home Green Resort Mooi Bemelen-31	230.80
19	29	Holiday Home Green Resort Mooi Bemelen-15	230.80
20	30	Holiday Home Green Resort Mooi Bemelen-10	311.63
21	31	Holiday Home Green Resort Mooi Bemelen-13	311.63
22	32	Holiday Home Green Resort Mooi Bemelen-16	311.63
23	33	Resort Mooi Bemelen	130.22
24	34	Holiday Home Green Resort Mooi Bemelen-6	230.80
25	35	Holiday Home Green Resort Mooi Bemelen-3	230.80

3.2. Agregarea datelor

Summary Statistics

❖ *Descrierea problemei*

Problema vizată prin agregarea datelor și folosirea Summary Statistics este nevoie de a obține o imagine de ansamblu asupra valorilor numerice, cum ar fi prețurile, pe grupuri relevante.

❖ *Informații necesare pentru rezolvare*

- Tabela inițială cu hoteluri și prețuri pentru a crea subsetul filtrat și tabela agregată;

- Coloana City, pentru filtrarea hotelurilor din zonele Amsterdam City Center, Bemelen și Delft;
- Coloana Price, pentru calculul statisticilor (media, suma, mediana);

❖ Produs software / funcție / metodă de calcul folosită

- Software: SAS Enterprise Guide

Functii si metode folosite:

- Filtrare cu opțiunea IN LIST pentru selectarea hotelurilor din zone specifice (Amsterdam City Center, Bemelen, Delft);
- Summary Statistics Wizard din meniul Tasks > Describe pentru calculul statisticilor agregate (media, suma totală, mediana);
- Crearea unei tabele agregate care grupează datele și salvează rezultatele statisticilor calculate.

❖ Rezolvarea cu ajutorul produsului software

Mai întâi, am creat o tabelă nouă prin aplicarea unui filtru pe coloana City, folosind opțiunea IN LIST pentru a include doar hotelurile din zonele specifice: Amsterdam City Center, Bemelen și Delft. Această filtrare asigură că analiza se face doar pe datele relevante pentru aceste locații. Ulterior, am utilizat Summary Statistics Wizard, accesând secțiunea Tasks și apoi Describe, pentru a genera o tabelă agregată care conține statistici descriptive. A m selectat variabila Price pentru analiză și am ales să calculez indicatori precum media (mean), suma totală (sum) și mediana (median) a prețurilor pentru fiecare grupă de hoteluri filtrate.

	ID	Name	Price	City	State
1	2	Multatuli Hotel	143.69	Amsterdam City Center	Amsterdam
2	6	Eden Hotel Amsterdam	220.66	Amsterdam City Center	Amsterdam
3	9	Andaz Amsterdam Prinsengracht - a concept by Hyatt	495.42	Amsterdam City Center	Amsterdam
4	10	Hyatt Regency Amsterdam	343.51	Amsterdam City Center	Amsterdam
5	11	NH Collection Amsterdam Flower Market	323.30	Amsterdam City Center	Amsterdam
6	12	ibis Styles Amsterdam Central Station	195.78	Amsterdam City Center	Amsterdam
7	14	Rho Hotel	206.56	Amsterdam City Center	Amsterdam
8	16	Hotel V Nesplein	241.09	Amsterdam City Center	Amsterdam
9	17	Kimpton De Witt Amsterdam, an IHG Hotel	380.79	Amsterdam City Center	Amsterdam
10	21	The White Tulip Hostel	134.71	Amsterdam City Center	Amsterdam

Summary Statistics Results			
The MEANS Procedure			
Analysis Variable : Price			
City	Mean	Sum	Median
Amsterdam City Center	259.09	3109.05	226.91
Bemelen	250.27	6256.78	230.80
Delft	124.49	2489.74	105.53

Rezultatele analizei arată că prețurile medii în Amsterdam City Center și Bemelen sunt relativ apropriate, situându-se în jurul valorilor de 259,09, respectiv 250,27, iar mediile acestora sunt semnificativ mai mari față de cele din Delft, unde media prețurilor este de 124,49. Mediana prețurilor confirmă această tendință, fiind mai ridicată în Amsterdam City Center și Bemelen (226,91 și, respectiv, 230,80) comparativ cu Delft (105,53). Suma totală a prețurilor este cea mai mare în Bemelen, ceea ce poate indica un număr mai mare de observații sau prețuri individuale mai ridicate. De asemenea, faptul că media este ușor mai mare decât mediana în toate cele trei orașe sugerează prezența unor valori extreme care influențează media, indicând o distribuție asimetrică a prețurilor spre valori mai mari. Astfel, se poate concluziona că prețurile sunt în general mai ridicate în Amsterdam City Center și Bemelen, în timp ce Delft se remarcă prin prețuri considerabil mai mici.

Summary Tables

❖ Descrierea problemei

Problema vizată în realizarea summary tables a fost de a sintetiza și compara valorile medii ale prețurilor și ratingurilor hotelurilor în funcție de orașe, pornind de la datele detaliate din tabela inițială. Scopul acestei analize a fost de a obține o imagine clară și structurată asupra performanței financiare și calitative a hotelurilor pe diferite regiuni, facilitând astfel identificarea zonelor cu cele mai bune sau cele mai slabe rezultate.

❖ Informații necesare pentru rezolvare

- Tabela inițială care conține datele hotelurilor, cu coloanele **Price** (preț), **Rating** (evaluare) și **City** (oraș);
- Variabilele de analizat, respectiv **Price** și **Rating**, pentru care se vor calcula statisticile descriptive;
- Variabila de clasificare **City**, după care se vor grupa datele pentru a calcula mediile pe fiecare oraș în parte;

❖ Produs software / funcție / metodă de calcul folosită

- **Software:** SAS Enterprise Guide

Functii si metode folosite:

- Describe → Summary Tables — opțiunea folosită pentru a crea tabele sumare (aggregate) cu statistici descriptive;
- Selectarea variabilelor de analiză (Analysis variables);
- Selectarea variabilelor de clasificare (Classification variables);
- Setarea variabilelor în zona Preview — plasarea variabilelor în poziții de rând sau coloană pentru configurarea structurii tabelului;
- Preview -> Box Area Properties — setarea etichetei și a stilului fontului și Table Properties — configurarea opțiunilor tabelului;
- Formatarea valorilor datelor (Data Value Properties);
- Adăugarea titlului raportului;

❖ Rezolvarea cu ajutorul produsului software

Pentru realizarea raportului, am început prin a deschide tabela inițială care conține date despre hoteluri, inclusiv prețurile, ratingurile și orașele acestora. Am accesat meniul Describe și am selectat opțiunea Summary Tables, unde am setat variabilele de analiză ca fiind prețul și ratingul, iar variabila de clasificare a fost orașul. Astfel, am creat un tabel sumarizat care afișează valorile medii ale prețurilor și ratingurilor grupate pe orașe. În final, am aplicat opțiuni de formatare pentru a personaliza aspectul tabelului, facilitând astfel o interpretare mai clară a datelor în funcție de regiuni.

Prețul și Ratingul după Regiune		
Regiune	Price	Rating
	Mean	Mean
Amsterdam	190.474	8.178
Eindhoven	92.409166667	8.5375
Maastricht	325.69	8
Rotterdam	127.0156	8.728
The Hague	136.64952381	8.7476190476
Utrecht	133.14434783	8.747826087
Eindhoven	68.28	8.1
Friesland	102.5975	8.255
Gelderland	145.25933333	8.24
Groningen	119.1628	8.632
Limburg	169.97585714	8.3342857143
Maastricht	200.76	8.65
North Brabant	114.54209302	8.288372093
North Holland	133.09169231	8.2430769231
Overijssel	106.53095238	8.280952381
South Holland	124.487	8.335
The Hague	138.44384615	7.8076923077
Utrecht	121.24	8.6
Zeeland	140.9884	8.192

Tabelul prezintă prețurile și ratingurile medii ale hotelurilor pe regiuni din Olanda, evidențiind variații importante. De exemplu, Maastricht are prețuri medii ridicate (325,69), iar Eindhoven și Friesland prețuri mult mai mici (92,41 și 102,59). Ratingurile medii sunt în general ridicate, între 8,1 și 8,86, cu regiuni ca Rotterdam și Maastricht peste 8,7, semnalând satisfacție mare a clientilor. Nu există o corelație directă între preț și rating, ceea ce sugerează că și alți factori influențează calitatea percepță, nu doar prețul.

4. Grafice

❖ Descrierea problemei

Problema vizată prin realizarea graficelor este reprezentarea vizuală clară și intuitivă a distribuției datelor referitoare la hoteluri pe regiuni și orașe. Graficul tip bar chart este folosit pentru a arăta prețul total pe fiecare regiune, facilitând compararea rapidă a volumului financiar generat în diferite zone. Pie chart-ul evidențiază numărul de hoteluri în funcție de oraș, oferind o imagine asupra ponderii fiecărui oraș în cadrul întregii oferte hoteliere. Scopul acestor grafice este de a simplifica înțelegerea datelor complexe și de a susține luarea deciziilor bazate pe o interpretare vizuală rapidă a informațiilor despre prețuri și distribuția hotelurilor.

❖ Informații necesare pentru rezolvare

Pentru realizarea graficelor am utilizat setul de date initial, și anume coloanele urmatoare:

- City - pentru a grupa hotelurile în funcție de locație;
- Regiunea – pentru agregarea valorilor totale și calculul mediei per regiune;
- Prețul (Price) – folosit pentru calculul totalului și mediei prețurilor;
- Ratingul (Rating) – utilizat pentru analiza calității serviciilor oferite;
- Numărul hotelurilor – determinat prin frecvența înregistrărilor pentru fiecare oraș;

❖ Produs software / funcție / metodă de calcul folosită

- Software: SAS Enterprise Guide

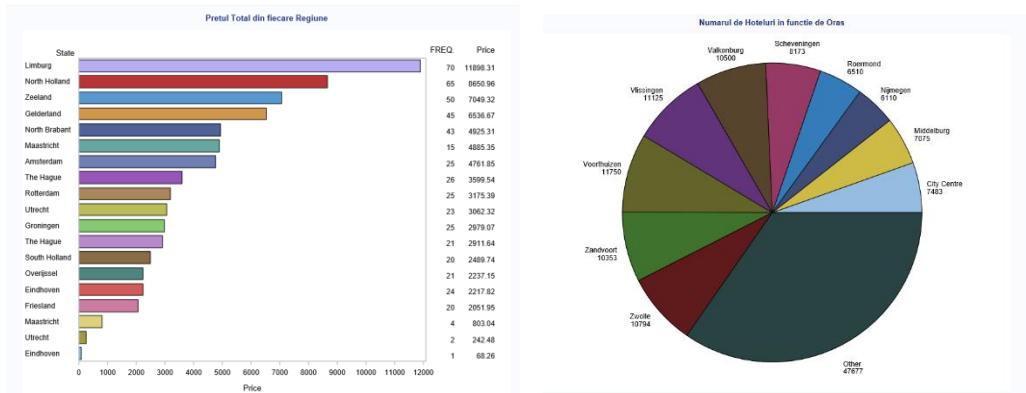
Functii si metode folosite:

- Graph -> Bar Chart/Pie Chart – pentru a crea un grafic de tip bară, respectiv placinta;
- În tab-ul Bar Chart am ales opțiunea Horizontal Colored Bar pentru o mai bună vizibilitate;
- În tab-ul Data, variabila CategoryName a fost setată ca Column to chart, iar Profit ca Sum of, pentru a reprezenta profitul total pe categorie;
- În tab-ul Appearance → Layout, s-a ales sortarea descrescătoare Descending bar height pentru a evidenția categoriile cu profit mai mare;
- În tab-ul Titles, s-a debifat opțiunea Use default text și s-a setat titlul Total Profit by Category;
- În Appearance → Advanced, s-a bifat opțiunea Specify one statistical value to show for bars și s-a ales Sum pentru a afișa totalul profitului per categorie;

❖ Rezolvarea cu ajutorul produsului software

Pornind de la tabela inițială din SAS, am accesat opțiunea Graph, iar pentru primul grafic am selectat Bar Chart. În cadrul acestuia, am reprezentat prețul total pe regiune, atribuind variabila city rolului *Column to chart* și variabila price rolului *Sum of*, astfel încât lungimea coloanelor să reflecte suma prețurilor. Pentru o afișare clară, am formatat valorile prețului cu două zecimale accesând *Properties → Change*, unde am modificat numărul de zecimale. În tab-ul *Appearance*, am selectat *Layout*, iar din lista *Order* am ales opțiunea *Descending bar height*, pentru a ordona coloanele în mod descrescător. Am adăugat un titlu sugestiv graficului, apoi am rulat procedura.

Pentru al doilea grafic, am urmat pași similari, dar am ales tipul Pie Chart. Acesta a fost folosit pentru a ilustra numărul de hoteluri în funcție de oraș, facilitând o comparație vizuală rapidă între localități.



Graficul bară arată prețul total al hotelurilor pe regiune, evidențierind că regiunile cu cele mai multe hoteluri, precum Limburg și North Holland, înregistrează și cele mai mari sume totale ale prețurilor. Regiunile cu puține hoteluri, precum Eindhoven sau Utrecht, au prețuri totale mult mai mici, ceea ce sugerează o ofertă redusă sau o acoperire limitată a datelor.

Graficul pie ilustrează distribuția hotelurilor pe orașe. Categoria „Other” deține cea mai mare parte, semnalând o dispersie mare în orașe nemenționate individual. Orașe precum

Voorrhuzen, Vlissingen și Zwolle au cele mai multe hoteluri, reflectând o concentrare semnificativă a unităților de cazare în aceste locații.

Realizarea de corelații și grafice de tip scatter

❖ Descrierea problemei

Problema vizată prin realizarea corelațiilor și a graficelor de tip scatter este identificarea relației dintre variabile cantitative, în special dintre prețul hotelurilor și evaluările acestora. Pentru a analiza această legătură, am efectuat o corelație între variabilele „rating” și „price”, utilizând și opțiunea de afișare grafică sub forma unui scatter plot, pentru a vizualiza modul în care valorile se distribuie și dacă există o tendință clară între ele.

Ulterior, am realizat un scatter plot cu linie de regresie pentru a evidenția mai clar relația dintre preț și scorul ajustat, observând dacă prețurile mai mari sunt asociate cu scoruri mai bune (sau invers). Aceste grafice ajută la detectarea unei posibile corelații liniare și oferă o bază vizuală pentru interpretarea legăturii dintre variabile.

❖ Informații necesare pentru rezolvare

Pentru realizarea corelațiilor și a graficelor de tip scatter, au fost necesare următoarele variabile din tabela de bază: Price, Rating și Scorul ajustat calculat anterior.

❖ Produs software / funcție / metodă de calcul folosită

- **Software:** SAS Enterprise Guide

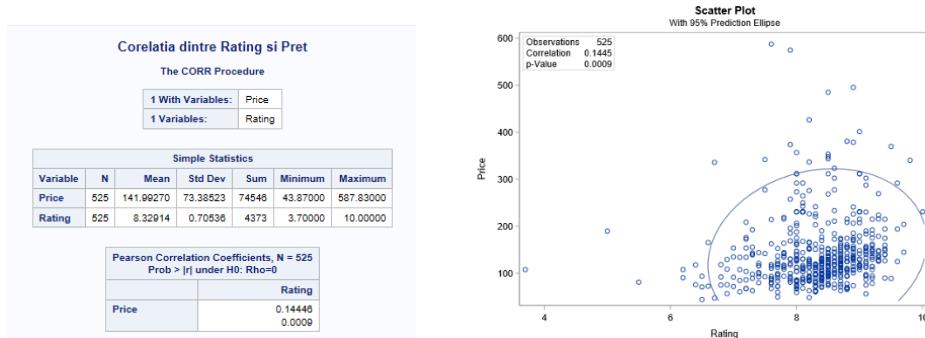
Functii si metode folosite:

- Analyse > Multivariate > Correlations – pentru a calcula coeficientul de corelație între variabilele Rating și Price.
- În fereastra de configurare, am alocat variabilele astfel: Analysis variables: Rating și Correlate with: Price;
- Am bifat opțiunea Display scatter plots pentru a genera automat o reprezentare grafică de tip scatter plot care să evidențieze vizual relația dintre cele două variabile;
- Graph > Scatter Plot with Regression Line – pentru a evidenția vizual legătura dintre Price și Adjusted Score;

❖ Rezolvarea cu ajutorul produsului software

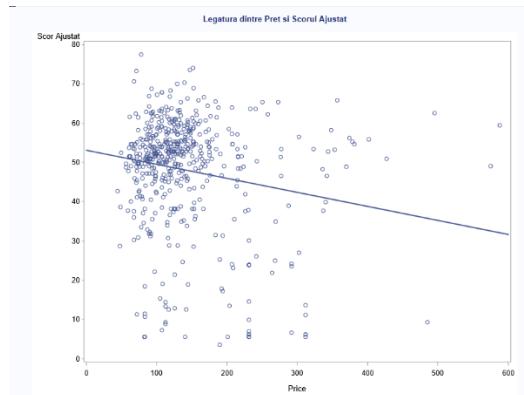
Am selectat tabela de date, apoi, din meniul Analyse, am accesat secțiunea Multivariate și am ales Correlations. În fereastra de configurare, am atribuit variabila Rating drept Analysis variables și variabila Price în Correlate with. Am activat opțiunea de afișare a graficului sub formă de scatter plot și am rulat analiza pentru a obține coeficientul numeric de corelație și reprezentarea grafică a relației dintre rating și preț. Ulterior, am creat un grafic scatter cu linie de regresie pentru a evidenția relația dintre preț și scorul ajustat. Acest grafic ne-a permis să vizualizăm clar tendința și intensitatea legăturii dintre cele două variabile.

Corelatia dintre rating si pret, respectiv graficul de tip scatter:



Analiza corelației dintre „Rating” și „Price” arată o legătură pozitivă, dar foarte slabă (coeficient Pearson 0,1445). Deși relația este statistic semnificativă ($p = 0,0009$), corelația este neglijabilă din punct de vedere practic. Media prețului este de aproximativ 142, iar media ratingului 8,33, cu o variabilitate mult mai mare la prețuri. Graficul scatter confirmă această legătură slabă, punctele fiind dispersate, cu o ușoară tendință pozitivă. Astfel, ratingul nu este un predictor puternic al prețului, indicând că și alți factori influențează mult mai mult prețul.

Graficul de tip scatter cu linie de regresie:



Graficul „Legătura dintre Preț și Scorul Ajustat” arată o relație liniară negativă slabă spre moderată: pe măsură ce prețul crește, scorul ajustat tinde să scadă. Deși linia de regresie indică această tendință, dispersia punctelor arată că prețul nu este un predictor precis al scorului ajustat. Majoritatea datelor se concentrează pe prețuri mici și scoruri mai mari, iar prezența unor valori aberante evidențiază variabilitatea și influența altor factori asupra scorului ajustat.

