FEAA
FACULTATEA DE ECONOMIE
ȘI ADMINISTRAREA AFACERILOR
UNIVERSITATEA "AL. I. CUZA" IAȘI

Proiect Tabelare și data visualisation

Student: Dancă Alexandra-Simona

310440105001SM211018

DM21

IAȘI 2022

FEAA
FACULTATEA DE ECONOMIE
ȘI ADMINISTRAREA AFACERILOR
UNIVERSITATEA "AL. I. CUZA" IAȘI

CUPRINS

1. Introducere

Sectorul imobiliar este o industrie importantă și există o cerere mare pentru o mai bună înțelegere a mecanismului vânzărilor în industrie și a factorilor care determină accelerarea acestor vânzări.

Deși această industrie imobiliară oferă informații valoroase pentru planificarea și dezvoltarea locuințelor pentru diferite grupuri rezidențiale cu nevoi diferite, devine importantă anticiparea celor mai esențiali factori care pot declanșa schimbarea prețurilor.

În general, prețul caselor este exprimat ca un grad de satisfacție a clienților, observat în valoarea pe care aceștia sunt dispuși să o plătească atunci când vor să o achiziționeze. Acest lucru se manifestă în cadrul datelor sub diferite forme, cum ar fi numărul de dormitoare și băi, suprafața terenului sau chiar importanța pe care o acordă clientul serviciilor și produselor de care locuința dispune.

Baza de date utilizată în această lucrare este obținută de pe Housing Prices Dataset. Aceasta conține 13 variabile, atât numerice, cât și nenumerice pentru 545 de observații.

Lucrarea are ca obiectiv principal, analiza factorilor care determină diferența de preț dintre case. Acest obiectiv va fi atins prin utilizarea unor metode statistice precum analiza de regresie, analiza de corelație și analiza descriptivă.

2. Prezentarea bazei de date

Baza de date inițială conține 13 factori pe care un client le poate lua în considerare atunci când dorește să achiziționeze o locuință.

```
# import librarii
from pandas import read_csv
import pandas as pd

# import baza de date
houses_df = read_csv('Housing.csv')
```

Figura 1. Input code

houses_df - DataFrame

Index	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished
5	10850000	7500	3	3	1	yes	no	yes	no	yes	2	yes	semi-furnished
6	10150000	8580	4	3	4	yes	no	no	no	yes	2	yes	semi-furnished
7	10150000	16200	5	3	2	yes	no	no	no	no	0	no	unfurnished
8	9870000	8100	4	1	2	yes	yes	yes	no	yes	2	yes	furnished
9	9800000	5750	3	2	4	yes	yes	no	no	yes	1	yes	unfurnished
10	9800000	13200	3	1	2	yes	no	yes	no	yes	2	yes	furnished
11	9681000	6000	4	3	2	yes	yes	yes	yes	no	2	no	semi-furnished
12	9310000	6550	4	2	2	yes	no	no	no	yes	1	yes	semi-furnished

Figura 2. Baza de date inițială

Variabilele din Figura 2 au următoarea semnificație:

- ***price*** → prețul unei case;
- ***area*** → suprafața terenului;
- ***bedrooms*** → numărul de dormitoare;
- ***bathrooms*** → numărul de băi;
- ***stories*** → numărul de magazie;
- ***mainroad*** → ieșire la drum principal;
- ***guestroom*** → camera de oaspeți;
- ***basement*** → subsol;
- ***hotwaterheating*** → încălzirea apei calde;
- ***airconditioning*** → aer conditionat;
- ***parking*** → numărul locurilor de parcare;

– ***prefarea*** → căsuță exterioară amenajată pentru relaxare;
– ***furnishingstatus*** → mobilier interior.

Pe parcursul acestui proiect, vom lucra cu 8 variabile. 7 dintre acestea sunt extrase din baza de date iar ultima este transformată din variabilă numerică în variabile categoriale.

2.1. Operații preliminare

Primul pas efectuat pentru operațiile peliminare a fost verificarea dacă în baza de date există valori lipsă.

```
9   ################## 2. PREZENTAREA BAZEI DE DATE ##################
10  ###################################################################
11  ######  OPERATII PRELIMINARE  ######
12  # verificam cate NA avem
13  houses_df.isnull().sum()
```

Figura 3. Input code

```
Out[6]:
price             0
area              0
bedrooms          0
bathrooms         0
stories           0
mainroad          0
guestroom         0
basement          0
hotwaterheating   0
airconditioning   0
parking           0
prefarea          0
furnishingstatus  0
dtype: int64
```

Figura 4. Output valori lipsă

Rezultatul afișează lipsa valorilor nule în baza de date inițiale, deci în continuare aceasta nu reprezintă o problemă pentru analiza dorită.

Înainte să eliminăm variabilele care nu o să ne fie de ajutor în proiect, s-a efectuat o serie de transformări. Cele 3 condiții de selecție a observațiilor dorite sunt:

– să nu avem camera de oaspeți;
– casa să aibă ieșire la drumul principal;
– să nu avem case cu mai mult de 2 locuri de parcare.

```
# Din baza initiala se va face o selectie care sa includa conditii pentru cel putin 2 variabile
# sa nu avem camera de oaspeti
houses_df1 = houses_df.loc[houses_df['guestroom'] == 'no']
# casa sa fie la drumul principal
houses_df2 = houses_df1.loc[houses_df['mainroad'] == 'yes']
# sa avem mai putin de 3 parcari
houses_df3 = houses_df2.loc[houses_df['parking'] < 3]
# dataset-ul final
houses_df_final = houses_df3
```

Figura 5. Input code

```
houses_df_final  DataFrame  (367, 13)  Column names: price, area, bedrooms, bathrooms, stories,
                                        mainroad, gue ...
```

Figura 6. DataFrame după selecții

În Figura 6 se poate observa că din 545 de observații, am mai rămas cu 367.

2.2. Transformarea variabilelor

Cu ajutorul funcției definită în Python, se va crea variabila categorială pe baza variabilei numerice.

```
######  TRANSFORMAREA VARIABILEI  ######
# definim noua variabila categoriala dintr-o variabila numerica
def function_num_to_cat(df):
    if df['stories'] == 1:
        return 'one'
    elif df['stories'] == 2:
        return 'two'
    else:
        return 'more'

# aplicam functia de mai sus
houses_df_final['stories_cat'] = houses_df_final.apply(lambda df: function_num_to_cat(df), axis=1)
```

Figura 7. Inpute code

Variabila categorială *stories_cat* derivă din variabila numerică *stories*. Astfel, numărul de magazie pe care o poate avea o casă, a fost împărțită în 3 categorii:

- stories = 1 → ”one”
- stories = 2 → ”two
- stories ≥ 3 → ”more”

```
# eliminam variabilele
houses_df_final = houses_df_final.drop(['stories','mainroad','guestroom','hotwaterheating','prefarea','furnishingstatus'], axis=1)
```

Figura 8.Input code

Mai departe s-a renunțat la variabilele *stories*, *mainroad*, *guestroom*, *hotwaterheating*, *prefarea* și *furnishingstatus* deoarece nu o să avem nevoie de acestea în analizele care urmează.

houses_df_final DataFrame (367, 8) Column names: price, area, bedrooms, bathrooms, basement, aircondition ...

houses_df_final - DataFrame

Index	price	area	bedrooms	bathrooms	basement	airconditioning	parking	stories_cat
0	13300000	7420	4	2	no	yes	2	more
2	12250000	9960	3	2	yes	no	2	two
5	10850000	7500	3	3	yes	yes	2	one
6	10150000	8580	4	3	no	yes	2	more
7	10150000	16200	5	3	no	no	0	two
10	9800000	13200	3	1	yes	yes	2	two
12	9310000	6550	4	2	no	yes	1	two
13	9240000	3500	4	2	no	no	2	two
14	9240000	7800	3	2	no	no	0	two
15	9100000	6000	4	1	yes	no	2	two
17	8960000	8500	3	2	no	yes	2	more
19	8855000	6420	3	2	no	yes	1	two

Figura 9. DataFrame final

Noua bază de date este compusă acum din 8 variabile. Aceasta este baza de date finală pe care se va face analiza pentru a îndeplini obiectivele propuse și este salvată într-un fisier Excel sub numele de ” Houses_FINAL.csv”.

```
42  ######  EXPORT BAZA DE DATE  ######
43  houses_df_final.to_csv('Houses_FINAL.csv')
44
45  ######  DESCRIEREA BAZEI DE DATE  ######
46  print(houses_df_final.info())
```

Figura 10. Input code

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 367 entries, 0 to 544
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   price            367 non-null    int64
 1   area             367 non-null    int64
 2   bedrooms         367 non-null    int64
 3   bathrooms        367 non-null    int64
 4   basement         367 non-null    object
 5   airconditioning  367 non-null    object
 6   parking          367 non-null    int64
 7   stories_cat      367 non-null    object
dtypes: int64(5), object(3)
memory usage: 25.8+ KB
None
```

Figura 11. Tipuri de date ale variabilelor

Se remarcă faptul că baza de date este alcătuită din 5 variabile de tip numeric și 3 variabile de tip categorial.

```
47    # analiza succinta pentru var numerice + nenumerice
48    describe_all = houses_df_final.describe(include='all')
```

Figura 12. Input code

describe_all - DataFrame

Index	price	area	bedrooms	bathrooms	basement	airconditioning	parking	stories_cat
count	367	367	367	367	367	367	367	367
unique	nan	nan	nan	nan	2	2	nan	3
top	nan	nan	nan	nan	no	no	nan	two
freq	nan	nan	nan	nan	268	257	nan	158
mean	4.72948e+06	5206.92	2.91281	1.26158	nan	nan	0.692098	nan
std	1.80706e+06	2287.08	0.730315	0.475889	nan	nan	0.823506	nan
min	1.75e+06	1700	1	1	nan	nan	0	nan
25%	3.5e+06	3610	2	1	nan	nan	0	nan
50%	4.305e+06	4500	3	1	nan	nan	0	nan
75%	5.74e+06	6323	3	1	nan	nan	1	nan
max	1.33e+07	16200	6	3	nan	nan	2	nan

Figura 13. Analiza descriptivă a variabilelor

În Figura 13 s-a realizat o analiză descriptivă concisă a celor 8 variabile din baza de date. Toate acestea vor fi analizate în profunzime în capitolul 3.

```
49    houses_df_final.nunique()
```

Figura 14. Input code

```
Out[23]:
price             170
area              208
bedrooms            6
bathrooms           3
basement            2
airconditioning     2
parking             3
stories_cat         3
dtype: int64
```

Figura 15. Valori unice ale variabilelor

Cele mai multe valori distincte se regasesc în coloana *area*. Acest lucru se întâmplă datorită faptului că probabilitatea ca mai multe case să aibă aceeași suprafață a terenului este mai mică față de probabilitatea ca mai multe locuințe să aibă același număr de dormitoare sau băi. O casă poate avea un singur subsol sau deloc.

3. Analiza grafică și numerică a varibilelor analizate

În acest capitol se analizează fiecare variabilă în parte. Acestă analiză calculează statistici, efectuează teste de ipoteze și construiește grafice.

3.1. Analiza descriptivă a variabilelor numerice și nenumerice

Analiza descriptivă a variabilelor numerice analizează media, mediana, minimul, maximul cât și quartilele, iar pentru variabilele nenumerice o analiza descriptivă a grupurilor.

3.1.1. Analiza descriptiva a variabilelor numerice

Pentru această analiză voi crea un subset de date, în care voi include doar cele 5 variabile numerice pentru a putea aplica aceeași funcție asupra tuturor variabilelor numerice .

```
51  ################## 3. ANALIZA GRAFICA SI NUMERICA A VARIABILELOR ANALIZATE ##################
52  ############################################################
53  ######  ANALIZA DESCRIPTIVA A VARIABILELOR NUMERICE  ######
54  houses_df_num = houses_df_final[['price','area','bedrooms','bathrooms','parking']]
55  # masuri de tendinta centrala si de variabilitate
56  print(houses_df_num.describe())
```

Figura 16. Input code

	price	area	bedrooms	bathrooms	parking
count	3.670000e+02	367.000000	367.000000	367.000000	367.000000
mean	4.729482e+06	5206.918256	2.912807	1.261580	0.692098
std	1.807060e+06	2287.080109	0.730315	0.475889	0.823506
min	1.750000e+06	1700.000000	1.000000	1.000000	0.000000
25%	3.500000e+06	3610.000000	2.000000	1.000000	0.000000
50%	4.305000e+06	4500.000000	3.000000	1.000000	0.000000
75%	5.740000e+06	6323.000000	3.000000	1.000000	1.000000
max	1.330000e+07	16200.000000	6.000000	3.000000	2.000000

Figura 17. Analiza descriptivă variabile numerice

A. Price

Prețul **mediu** al unei case este egal cu 4.729.482$ iar unitățile se **abat de la medie** cu 1.807.060$.

Prețul **minim** este egal cu 1.750.000$. Prețul **maxim** este egal cu 13.300.000$.

Q1 (quartila 1) - ne arată faptul că 25% dintre case au prețul de până la 3.500.000$, iar restul de 75% peste 3.500.000$.

Q2 (quartila2/mediana) - ne arată faptul că 50% dintre case au prețul de până la 4.305.000$, iar restul de 50% peste 4.305.000$.

Q3 (quartila3) - ne arată faptul că 75% dintre case au prețul de până la 5.740.000$, iar restul de 25% peste 5.740.000$.

B. Area

Suprafața terenului este în **medie** egal cu 5206,91 m^2 iar unitățile se **abat de la medie** cu 2287,08 m^2.

Suprafața **maximă** a terenului este de 16200 m^2. Suprafața **minimă** a terenului este de 1700 m^2.

Q1 (quartila 1) - ne arată faptul că 25% dintre case au suprafața de mai puțin de 3610 m^2, iar restul de 75% de peste 3610 m^2 .

Q2 (quartila2/mediana) - ne arată faptul că 50% dintre case au suprafața de mai puțin de 4500 m^2, iar restul de 50% de peste 4500 m^2.

Q3 (quartila3) - ne arată faptul că 75% dintre case au suprafața de mai puțin de 6323 m^2, iar restul de 25% de peste 6323 m^2.

C. Bedrooms

O casă are în **medie** 3 dormitoare iar unitățile se **abat de la medie** cu 0.73, adica aproximativ 1 dormitor.

Casa care are numărul **maxim** de dormitoare are 6 dormitoare. Casa care are numărul **minim** de dormitoare are 1 dormitor.

Q1 (quartila 1) - ne arată faptul că 25% dintre case au de de până la 2 dormitoare, iar restul de 75% peste 2 dormitoare.

Q2 (quartila2/mediana) - ne arată faptul că 50% dintre case au de de până la 3 dormitoare, iar restul de 50% peste 3 dormitoare.

Q3 (quartila3) - ne arată faptul că 75% dintre case au de de până la 3 dormitoare, iar restul de 25% peste 3 dormitoare.

D. Bathrooms

O casă are în **medie** o baie iar unitățile se **abat de la medie** cu 0.47, adica aproximativ 0 băi.

Casa care are numărul **maxim** de băi are 3 băi. Casa care are numărul **minim** de băi are o baie.

Q1 (quartila 1) - ne arată faptul că 25% dintre case au de de până la o baie, iar restul de 75% peste.

Q2 (quartila2/mediana) - ne arată faptul că 50% dintre case au de de până la o baie, iar restul de 50% peste.

Q3 (quartila3) - ne arată faptul că 75% dintre case au de de până la o baie, iar restul de 25% peste.

E. Parking

O casă are în **medie** o parcare iar unitățile se **abat de la medie** cu 0.83, adica aproximativ o parcare.

Casa care are numărul **maxim** de parcări are 2. Casa care are numărul **minim** de parcări are 0 parcări.

Q1 (quartila 1) - ne arată faptul că 25% dintre case nu au parcare, iar restul de 75% au peste 0 parcări.

Q2 (quartila2/mediana) - ne arată faptul că 50% dintre case nu au parcare, iar restul de 50% au peste 0 parcări.

Q3 (quartila3) - ne arată faptul că 75% dintre case au de de până la o parcare, iar restul de 25% peste.

Coeficientul de asimetrie (Skewness)

```
57    # masuri de distributie: asimetria (skewness)
58    #from scipy.stats import skew
59    houses_df_num.skew(axis = 0, skipna = True)  # print(skew(houses_df_num))
```

Figura 18. Input code

```
Out[30]:
price         1.229081
area          1.549935
bedrooms      0.601648
bathrooms     1.539239
parking       0.624652
dtype: float64
```

Figura 19. Output asimetrie

Din Figura 20 se observă că distribuția celor cinci variabile este asimetrică la dreapta deoarece toate au valori mai mari decât 0.

Coeficientul de boltire (Kurtosis)

```
# masuri de distributie: boltirea
#from scipy.stats import kurtosis
houses_df_num.kurtosis(axis = 0, skipna = True)  # print(kurtosis(houses_df_num))
```

Figura 20. Input code

```
Out[31]:
price          2.068254
area           3.235132
bedrooms       1.103444
bathrooms      1.365340
parking       -1.243760
dtype: float64
```

Figura 21. Output boltire

Din Figura 21 se observă că distribuția variabilei *parking* este platicurtică, deoarece valoarea aferentă boltirii este mai mică de 0. Distribuția celorlalte variabile este leptocurtică deoarece valoarea aferentă boltirii este mai mare de 0.

3.1.2. Analiza descriptivă a variabilelor categoriale

```
######  ANALIZA DESCRIPTIVA A VARIABILELOR NENUMERICE  ######
houses_df_nenum= houses_df_final[['basement','airconditioning', 'stories_cat']]

for col in houses_df_nenum:
    print("******************")
    print("*** "+ col+ " ***")
    print("******************")
    print("Numar")
    print("******************")
    print(houses_df_nenum[col].value_counts())
    print("******************")
    print("%")
    print("******************")
    print(houses_df_nenum[col].value_counts(normalize =True))
    print("_____________________________________________________")
```

Figura 22. Input code

```
******************
*** basement ***
******************
Numar
******************
no      268
yes      99
Name: basement, dtype: int64
******************
%
******************
no      0.730245
yes     0.269755
Name: basement, dtype: float64
```

```
******************
*** airconditioning ***
******************
Numar
******************
no      257
yes     110
Name: airconditioning, dtype: int64
******************
%
******************
no      0.700272
yes     0.299728
Name: airconditioning, dtype: float64
```

```
******************
*** stories_cat ***
******************
Numar
******************
two     158
one     150
more     59
Name: stories_cat, dtype: int64
******************
%
******************
two     0.430518
one     0.408719
more    0.160763
Name: stories_cat, dtype: float64
```

Figura 23. Output analiza descriptiva variabile categoriale

Pentru variabila *basment* predomină cu 73% categoria ”no” față de ”yes”. Asta înseamnă ca majoritatea caselor nu au subsol.

Pentru varibila *stories_cat* observațiile din baza de date sunt predominate de categoria ”two” cu 43% față de ”one” cu 41% și ”more” cu 16%.

Variabila *airconditioning* este alcătuită din 70% răspunsuri de ”no” și 30% de ”yes”.

3.2.Analiza grafică a variabilelor numerice și nenumerice

În acest subcapitol se va realiza analiză grafică a variabilelor analizate și în capitolul anterior.

3.2.1. Analiza grafică a variabilelor numerice

```
######  ANALIZA GRAFICA A VARIABILELOR NUMERICE  ######
#import matplotlib.pyplot as plt
print(houses_df_num.hist('price', bins= 10 , align='right', color='green', edgecolor='black'))
print(houses_df_num.hist('area', bins= 10 , align='right', color='pink', edgecolor='black'))
print(houses_df_num.hist('bedrooms', bins= 5 , align='right', color='red', edgecolor='black'))
print(houses_df_num.hist('bathrooms', bins= 3 , align='right', color='yellow', edgecolor='black'))
print(houses_df_num.hist('parking', bins= 3 , align='right', color='blue', edgecolor='black'))
```

Figura 24. Input code

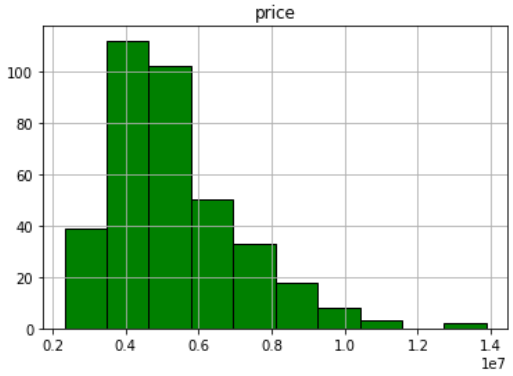


Figura 25. Histogramă variabilă price

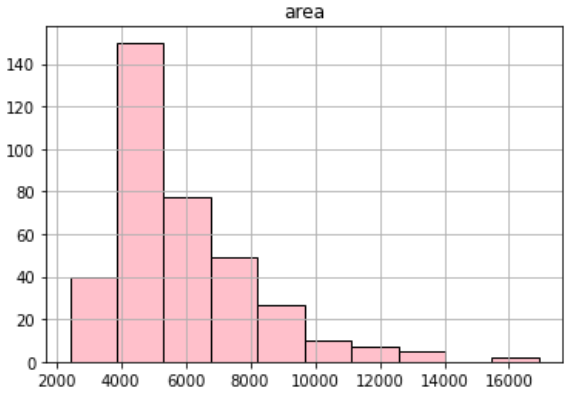


Figura 26. Histogramă variabilă area

Figura 27.Histogramă variabilă bedrooms

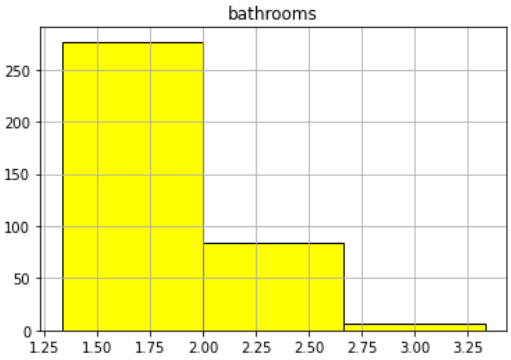


Figura 28. Histogramă variabilă bathrooms

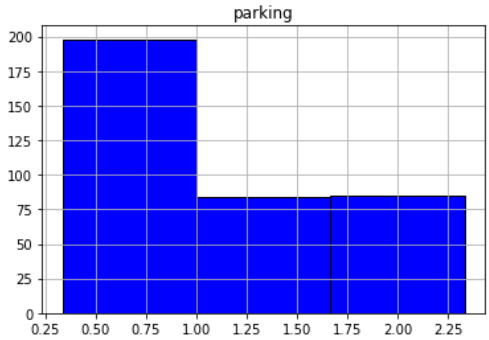


Figura 29. Histogramă variabilă parking

Se poate observa că histogramele descriu grafic tot ce a fost prezentat la subcapitolul anterior, Analiza descriptiva a variabilelor numerice.

3.2.2. Analiză grafică variabile nenumerice

```
88    ######  ANALIZA GRAFICA A VARIABILELOR NENUMERICE  ######
89    print(houses_df_nenum.basement.hist(bins=3, color='red'))
90    print(houses_df_nenum.airconditioning.hist(bins=3, color='blue'))
91    print(houses_df_nenum.stories_cat.hist(bins=5, color='orange'))
```

Figura 30. Input code

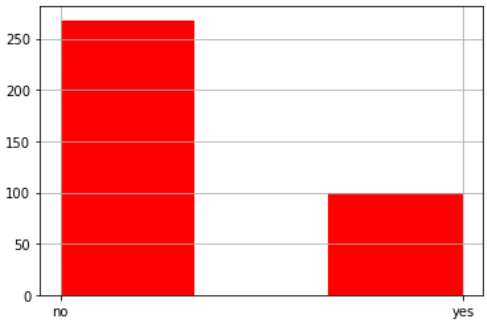


Figura 31. Histogramă variabilă basement

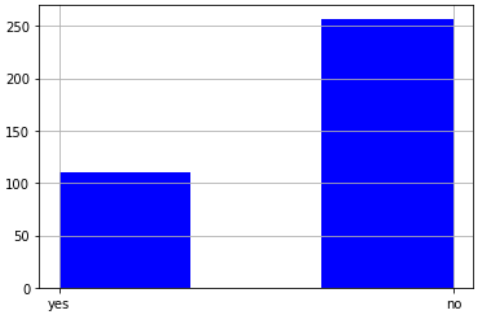


Figura 32. Histogramă variabilă airconditioning

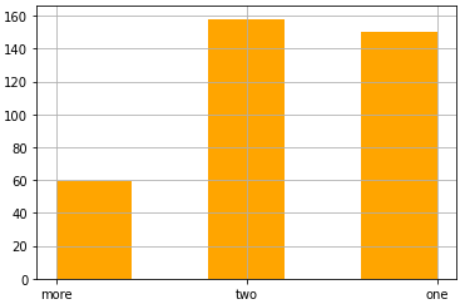


Figura 33. Histogramă variabilă stories_cat

Se poate observa cu ușurină din graficele de mai sus că observațiile analizate sunt alcătuite preponderent din case fără subsol, fără aer conditionat și cu 2 magazii.

3.3.Identificarea outlierilor și tratarea acestora

Pentru a identifica valorile extreme pentru variabilele studiate se va folosi graficul Box plot.

```
93    ######  IDENTIFICAREA OUTLIERILOR SI TRATAREA ACESTORA  ######
94    houses_df_num.boxplot('price')
95    houses_df_num.boxplot('area')
96    houses_df_num.boxplot('bedrooms')
97    houses_df_num.boxplot('bathrooms')
98    houses_df_num.boxplot('parking')
```

Figura 34. Input code

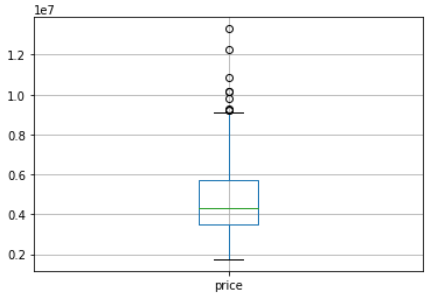


Figura 35. Box plot variabila price

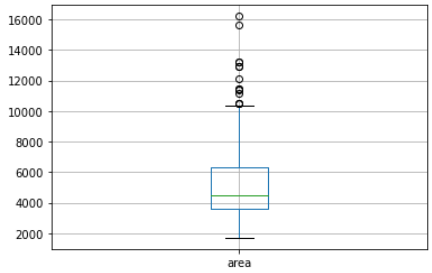
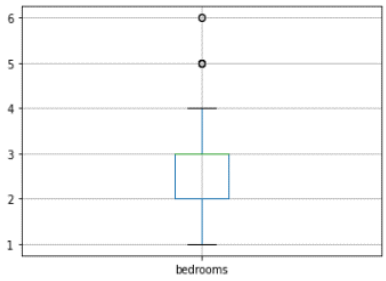
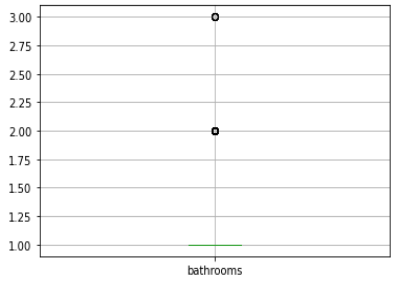


Figura 36. Box plot variabila area





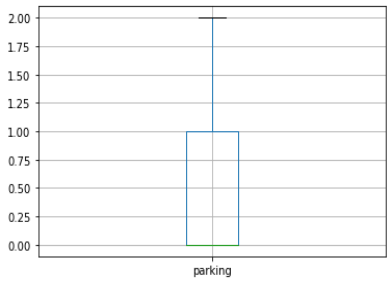


Figura 37. Box plot variabila bedrooms *bathrooms* *parking*

Putem observa că în toate diagramele box-plot, înafară de ultima, avem valori extreme, dar aceste nu vor fi eliminate deoarece nu afecteaza cu nimic analizele care urmează.

4. Analiza statistică a variabilelor categoriale

În acest capitol se vor tabela datele variabilelor categoriale, se va realiza analiza de asociere și analiza de concordanță.

4.1. Tabelarea datelor

A. Basement și airconditioning

```
101  ################## 4. ANALIZA STATISTICA A VARIABILELOR CATEGORIALE ##################
102  ############################################################
103  ######  TABELAREA DATELOR  ######
104  cross_table_basement_airconditioning = pd.crosstab(houses_df_nenum.basement, houses_df_nenum.airconditioning, margins=True, margins_name="Total" )
105  print(cross_table_basement_airconditioning)
```

Figura 38. Input code

airconditioning	no	yes	Total
basement			
no	190	78	268
yes	67	32	99
Total	257	110	367

Figura 39. Tabelarea datelor

Din tabel se observă că 257 de case nu au aer condiționat, iar 110 au. Din cele 257 de case, 190 nu au subsol. Din cele 110 de case care au aer conditionat, 32 au subsol iar restul nu au.

Se remarcă faptul că domină categoria cu case care nu au subsol și care nu au aer condiționat.

B. Basement și stories_cat

```
107  cross_table_basement_stories_cat = pd.crosstab(houses_df_nenum.basement, houses_df_nenum.stories_cat, margins=True, margins_name="Total" )
108  print(cross_table_basement_stories_cat)
```

Figura 40. Input code

stories_cat	more	one	two	Total
basement				
no	54	107	107	268
yes	5	43	51	99
Total	59	150	158	367

Figura 41. Tabelarea datelor

268 din case nu au subsol. Din acestea, 107 au o magazie, 107 au 2 magazii și 54 au mai mult de 2 magazii.

99 din case au subsol. Din acestea, 51 au 2 magazii, 43 au o magazie iar restul de 5 case au mai mult de 2 magazii.

C. Stories_cat și airconditioning

```
110  cross_table_airconditioning_stories_cat =pd.crosstab(houses_df_nenum.airconditioning,houses_df_nenum.stories_cat,margins=True,margins_name="Total" )
111  print(cross_table_airconditioning_stories_cat)
```

Figura 42. Input code

stories_cat airconditioning	more	one	two	Total
no	23	117	117	257
yes	36	33	41	110
Total	59	150	158	367

Figura 43. Tabelarea datelor

Cele mai multe dintre case, mai exact 257, nu au aer condiționat. Din acestea, la un număr egal de 117 fiecare, au una sau două magazii. Restul au mai mult de două magazii.

Dintre cele 110 de case cu aer condiționat, 33 dețin o magazie, 41 dețin două magazii și 36 au mai mult de două magazii.

4.2. Analiza de asociere

Cu ajutorul testului Chi-square verificăm dacă există o asociere semnificativă între categoriile celor două variabile pe care le vom alege.

A. Basement și airconditioning

```
113  ######  ANALIZA DE ASOCIERE  ######
114  from scipy.stats import chi2_contingency
115  chi_test_basement_airconditioning=chi2_contingency(cross_table_basement_airconditioning)
116  chi_test_basement_airconditioning
```

Figura 44. Input code

```
Out[57]:
(0.3568502526012466,
 0.9858547792103018,
 4,
 array([[187.67302452,  80.32697548, 268.        ],
        [ 69.32697548,  29.67302452,  99.        ],
        [257.        , 110.        , 367.        ]]))
```

Figura 45. Tabel analiza de asociere basement și airconditioning

1. Formularea ipotezelor

- H0: intre categoriile celor doua variabile nu exista o asociere semnificativa (variabilele sunt independente)
- H1: intre categoriile celor doua variabile exista o asociere semnificativa (variabilele nu sunt independente)

2. Regula de decizie

- $P_{value} < 0,05$, se respinge H0, cu un risc asumat de 5%
- $P_{value} \geq 0,05$, nu se respinge H0, cu o probabilitate de 95%

3. Decizie

P-value = 0,98 > α = 0,05, nu se respinge H0

4. Interpretare

Cu o probabilitate de 95%, nu există o asociere semnificativă între categoriile variabilei *basement* și *airconditioning*.

B. Basement și stories_cat

```
118  chi_test_basement_stories_cat=chi2_contingency(cross_table_basement_stories_cat)
119  chi_test_basement_stories_cat
```

Figura 46. Input code

```
Out[59]:
(12.72519013584575,
 0.047613550222115726,
 6,
 array([[ 43.08446866, 109.53678474, 115.37874659, 268.        ],
        [ 15.91553134,  40.46321526,  42.62125341,  99.        ],
        [ 59.        , 150.        , 158.        , 367.        ]]))
```

Figura 47. Tabel analiza de asociere basement și stories_cat

1. Formularea ipotezelor

- H0: intre categoriile celor doua variabile nu exista o asociere semnificativa (variabilele sunt independente)
- H1: intre categoriile celor doua variabile exista o asociere semnificativa (variabilele nu sunt independente)

2. Regula de decizie

- $P_{value} < 0,05$, se respinge H0, cu un risc asumat de 5%
- $P_{value} \geq 0,05$, nu se respinge H0, cu o probabilitate de 95%

3. Decizie

P_{value} = 0,047< α = 0,05, se respinge H0

4. Interpretare

Cu un risc asumat de 5% se respinge ipoteza nulă, deci variabilele sunt dependente.

C. Airconditioning și stories_cat

```
121  chi_test_airconditioning_stories_cat=chi2_contingency(cross_table_airconditioning_stories_cat)
122  chi_test_airconditioning_stories_cat
```

Figura 48. Input code

```
Out[61]:
(32.85189514479671,
 1.1196756993935817e-05,
 6,
 array([[ 41.31607629, 105.04087193, 110.64305177, 257.        ],
        [ 17.68392371,  44.95912807,  47.35694823, 110.        ],
        [ 59.        , 150.        , 158.        , 367.        ]]))
```

Figura 49. Tabel analiza de asociere airconditioning și stories_cat

1. Formularea ipotezelor

- H0: intre categoriile celor doua variabile nu exista o asociere semnificativa (variabilele sunt independente)
- H1: intre categoriile celor doua variabile exista o asociere semnificativa (variabilele nu sunt independente)

2. Regula de decizie

- $P_{value} < 0,05$, se respinge H0, cu un risc asumat de 5%
- $P_{value} \geq 0,05$, nu se respinge H0, cu o probabilitate de 95%

3. Decizie

P-value = 0,000011 < α = 0,05, se respinge H0

4. Interpretare

Cu un risc asumat de 5%, se respinge ipoteza nulă, deci variabilele nu sunt independente.

4.3.Analiza de concordanță

Testul de concordanță verifică dacă între distribuția empirică a unei variabile categoriale și o distribuție teoretică există diferențe semnificative

A. Basment

```
125  from scipy.stats import chisquare
126  chisquare(f_obs=houses_df_final['basement'].value_counts(),f_exp=None)
```

Figura 50. Input code

```
Out[63]: Power_divergenceResult(statistic=77.82288828337875,
pvalue=1.1270656361897891e-18)
```

Figura 51. Analiza de concordanță variabila basment

1. Formularea ipotezelor

- H0: între cele două distribuții nu există diferențe semnificative (există concordanță de structură)
- H1: cele două distribuții diferă semnificativ (nu există concordanță de structură)

2. Testul folosit: Chi-square (Chi patrat)

3. Regula de decizie

- P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%
- P_{value} ≥ 0,05, nu se respinge H0, cu o probabilitate de 95%

4. Decizie

P_{value} = 0,00000000000000000011< α = 0,05, se respinge H0

5. Interpretare

Cu un risc asumat de 5%, se respinge ipoteza nulă. Așadar, există diferențe semnificative între categoriile variabilei *besment.*

B. Airconditioning

```
127    chisquare(f_obs=houses_df_final['airconditioning'].value_counts(),f_exp=None)
```

Figura 52. Input code

```
Out[64]: Power_divergenceResult(statistic=58.880108991825615,
pvalue=1.6757774210307123e-14)
```

Figura 53. Analiza de concordanță variabila airconditioning

1. **Formularea ipotezelor**
   - H0: între cele două distribuții nu există diferențe semnificative (există concordanță de structură)
   - H1: cele două distribuții diferă semnificativ (nu există concordanță de structură)

2. Testul folosit: Chi-square (Chi patrat)

3. Regula de decizie

- P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%
- P_{value} > 0,05, nu se respinge H0, cu o probabilitate de 95%

4. Decizie

P_{value} = 0,000000000000016< α = 0,05, se respinge H0

5. Interpretare

Cu un risc asumat de 5% se respinge ipoteza nulă. Așadar, există diferențe semnificative între categoriile variabilei *airconditioning.*

C. Stories_cat

```
128    chisquare(f_obs=houses_df_final['stories_cat'].value_counts(),f_exp=None)
```

Figura 54. Input code

```
Out[65]: Power_divergenceResult(statistic=49.444141689373296,
pvalue=1.833753901807522e-11)
```

Figura 55. Analiza de concordanță variabila stories_cat

1. Formularea ipotezelor

- H0: între cele două distribuții nu există diferențe semnificative (există concordanță de structură)
- H1: cele două distribuții diferă semnificativ (nu există concordanță de structură)

2. Testul folosit: Chi-square (Chi patrat)

3. Regula de decizie

- P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%
- P_{value} > 0,05, nu se respinge H0, cu o probabilitate de 95%

4. Decizie

P_{value} = 0,000000000018 < α =0,05, se respinge H0

5. Interpretare

Cu un risc asumat de 5% se respinge ipoteza nulă. Așadar, există diferențe semnificative între categoriile variabilei *stories_cat.*

5. Estimarea și testarea mediilor

În acest capitol dorim să generăm o estimare a intervalului de încredere de 95% pentru o medie al celor 5 variabile analizate, iar apoi sa facem testarea mediilor populațiilor.

5.1. Estimarea mediei prin interval de încredere

Pentru estimarea mediei prin interval de încredere o sa ne folosim de o funcție definită de noi pentru a o putea folosi la toate cele 5 variabile numerice *price, area, bedrooms, bathrooms* și *parking*.

```
131  ################## 5. ESTIMAREA SI TESTAREA MEDIILOR ##################
132  ######################################################################
133  ######  ESTIMAREA MEDIEI PRIN IC  ######
134  import numpy as np
135  import scipy as sp
136  import scipy.stats
137  def interval_de_incredere (data, incredere):
138      a=1.0*np.array(data)
139      n=len(a)
140      media=np.mean(a)
141      sem=scipy.stats.sem(a)
142      h = sem*sp.stats.t.ppf((1+incredere)/2., n-1)
143      return media-h, media+h
144
145  print('Interval de incredere al variabilei price este:', interval_de_incredere(houses_df_final['price'], 0.95))
146  print('Interval de incredere al variabilei area este:', interval_de_incredere(houses_df_final['area'], 0.95))
147  print('Interval de incredere al variabilei bedrooms este:', interval_de_incredere(houses_df_final['bedrooms'], 0.95))
148  print('Interval de incredere al variabilei bathrooms este:', interval_de_incredere(houses_df_final['bathrooms'], 0.95))
149  print('Interval de incredere al variabilei parking este:', interval_de_incredere(houses_df_final['parking'], 0.95))
```

Figura 56. Input code

A. *Interval de încredere price*

```
Interval de incredere al variabilei price este: (4543989.2624437595, 4914974.170798748)
```

Figura 57. Interval de încredere price

Cu o probabilitate de 95% se poate afirma că intervalul de încredere al variabilei *price* este acoperit de valorile [4543989,26 , 4914974,17]$.

B. *Interval de încredere area*

```
Interval de incredere al variabilei area este: (4972.152393579564, 5441.684118682016)
```

Figura 58. Interval de încredere area

Cu o probabilitate de 95% se poate afirma că intervalul de încredere al variabilei *area* este acoperit de valorile [4972,15 , 5441,68] m^2.

C. *Interval de încredere bedrooms*

```
Interval de incredere al variabilei bedrooms este: (2.8378406246763577, 2.987772454342716)
```

Figura 59. Interval de încredere bedrooms

Cu o probabilitate de 95% se poate afirma că intervalul de încredere al variabilei *bedrooms* este acoperit de valorile [2,84 , 2,99] de camere.

D. *Interval de încredere bathrooms*

```
Interval de incredere al variabilei bathrooms este: (1.2127309403519493, 1.3104298225908302)
```

Figura 60. Interval de încredere bathrooms

Cu o probabilitate de 95% se poate afirma că intervalul de încredere al variabilei *bathrooms* este acoperit de valorile [1,21 , 1,31] de camere.

E. *Interval de încredere parking*

```
Interval de incredere al variabilei parking este: (0.6075662933411583, 0.7766298919449452)
```

Figura 61. Interval de încredere parking

Cu o probabilitate de 95% se poate afirma că intervalul de încredere al variabilei *parking* este acoperit de valorile [0,61 , 0,78] locuri de parcare.

5.2. Testarea unei medii cu o valoare fixă

A. Testarea mediei cu o valoare fixa a variabilei *price*

```
151  ######  TESTAREA MEDIILOR POPULATIEI  ######
152  ## testarea unei medii cu o valoare fixa
153  from scipy import stats
154  print(stats.ttest_1samp(houses_df_final.price, 4600000))
```

Figura 62. Input code

```
Ttest_1sampResult(statistic=1.372678409128741, pvalue=0.17069286022760044)
```

Figura 63. Output pentru price

1. Formularea ipotezelor

- H0: $\boldsymbol{\mu}$ = 4600000$ (în medie, prețul unei case nu diferă semnificativ de 4600000$)
- H1: $\boldsymbol{\mu} \neq$ 4600000$ (în medie, prețul unei case diferă semnificativ de 4600000$)

2. Regula de decizie

- P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%
- $P_{value} \geq$ 0,05, nu se respinge H0, cu o probabilitate de 95%

3. Decizie

P_{value} = 0,17 > α = 0,05, nu se respinge H0

4. Interpretare

Cu o probabilitate de 95%, media variabilei *price* nu diferă semnificativ de 4600000$.

B. Testarea mediei cu o valoare fixa a variabilei *area*

```
155    print(stats.ttest_1samp(houses_df_final.area, 3000)) # am dat o valoare inafara iC
```

Figura 64. Input cod

```
Ttest_1sampResult(statistic=18.48578521068327, pvalue=2.3369749121605574e-54)
```

Figura 65. Output pentru area

1. Formularea ipotezelor

- H0: $\boldsymbol{\mu}$ = 3000 m^2 (în medie, suprafața terenului nu diferă semnificativ de 3000 m^2)
- H1: $\boldsymbol{\mu} \neq$ 6 ani (în medie, suprafața terenului diferă semnificativ de 3000 m^2)

2. Regula de decizie

- P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%
- $P_{value} \geq$ 0,05, nu se respinge H0, cu o probabilitate de 95%

3. Decizie

P_{value} = 0,00.. < α = 0,05, se respinge H0

4. Interpretare

Cu un risc asumat de 5%, media variabilei *area* diferă semnificativ de 3000 m^2.

C. Testarea mediei cu o valoare fixa a variabilei *bedrooms*

```
156    print(stats.ttest_1samp(houses_df_final.bedrooms, 2.90))
```

Figura 66. Input cod

```
Ttest_1sampResult(statistic=0.3359344554350702, pvalue=0.7371127902195043)
```

Figura 67. Output pentru bedrooms

1. Formularea ipotezelor

- H0: $\boldsymbol{\mu}$ = 2,9 camere (în medie, numărul de dormitoare dintr-o casa nu diferă semnificativ de 2,9 camere)
- H1: $\boldsymbol{\mu} \neq$ 2,9 camere (în medie, numărul de dormitoare dintr-o casa diferă semnificativ de 2,9 camere)

2. Regula de decizie

- P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%

– $P_{value} \geq 0,05$, nu se respinge H0, cu o probabilitate de 95%

3. Decizie

$P_{value} = 0,73 > \alpha = 0,05$, nu se respinge H0

4. Interpretare

Cu o probabilitate de 95%, media variabilei *bedrooms* nu diferă semnificativ de 2,9 camere.

D. Testarea mediei cu o valoare fixa a variabilei *bathrooms*

```
157    print(stats.ttest_1samp(houses_df_final.bathrooms, 2)) # am dat o valoare inafara iC
```

Figura 68. Input cod

```
Ttest_1sampResult(statistic=-29.725572571344692, pvalue=1.2647371572097298e-99)
```

Figura 69. Output pentru bathrooms

1. Formularea ipotezelor

– H0: $\boldsymbol{\mu} = 2$ camere (în medie, numărul de băi dintr-o casa nu diferă semnificativ de 2,9 camere)

– H1: $\boldsymbol{\mu} \neq 2$ camere (în medie, numărul de băi dintr-o casa diferă semnificativ de 2,9 camere)

2. Regula de decizie

– $P_{value} < 0,05$, se respinge H0, cu un risc asumat de 5%

– $P_{value} \geq 0,05$, nu se respinge H0, cu o probabilitate de 95%

3. Decizie

$P_{value} = 0,00... < \alpha = 0,05$, se respinge H0

4. Interpretare

Cu un risc asumat de 5%, media variabilei *bathrooms* diferă semnificativ de 2 camere.

E. Testarea mediei cu o valoare fixa a variabilei *parking*

```
158    print(stats.ttest_1samp(houses_df_final.parking, 0.65))
```

Figura 70. Input cod

```
Ttest_1sampResult(statistic=0.9793296599181578, pvalue=0.3280640030705555)
```

Figura 71. Output pentru parking

1. Formularea ipotezelor

- H0: $\boldsymbol{\mu}$ = 0,65 locuri de parcare (în medie, numărul de locuri de parcare nu diferă semnificativ de 0,65 locuri)
- H1: $\boldsymbol{\mu} \neq$ 0,65 locuri de parcare (numărul de locuri de parcare diferă semnificativ de 0,65 locuri)

2. Regula de decizie

- P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%
- $P_{value} \geq$ 0,05, nu se respinge H0, cu o probabilitate de 95%

3. Decizie

P_{value} = 0,32 > α = 0,05, nu se respinge H0

4. Interpretare

Cu o probabilitate de 95%, media variabilei *parking* nu diferă semnificativ de 0,65 locuri de parcare.

5.3. Testarea diferenței dintre două medii – eșantioane independente

```
160  ## Testarea diferenței dintre 2 medii (esantioane independente sau esantioane perechi)
161  stories_one = houses_df_final.loc[houses_df_final['stories_cat']=='one']
162  stories_two = houses_df_final.loc[houses_df_final['stories_cat']=='two']
163  print(stats.ttest_ind(stories_one.area, stories_two.area))
```

Figura 72. Input code

```
Ttest_indResult(statistic=2.9654484856330208, pvalue=0.0032602673283195216)
```

Figura 73. Output testarea diferentei dintre mediile categoriilor de stories_cat ”one” și ”two” in functie de variabila are

1. Formularea ipotezelor

- H0: $\mu 1 = \mu 2$ (nu există diferențe semnificative între suprafața medie a terenului casei în funcție de categoriile ”one” și ”two” ale variabilei *store_cat*)
- H1: $\mu 1 \neq \mu 2$ (există diferențe semnificative între suprafața medie a terenului casei în funcție de categoriile ”one” și ”two” ale variabilei *store_cat*)

2. Regula de decizie

- P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%
- $P_{value} \geq$ 0,05, nu se respinge H0, cu o probabilitate de 95%

3. Decizie

P_{value} = 0,003 < α = 0,05, se respinge H0

4. Interpretare

Cu un risc asumat de 5%, putem spune ca există diferențe semnificative între suprafața medie a terenului casei în funcție de categoriile ”one” și ”two” ale variabilei *store_cat*.

5.4. Testarea diferenței dintre trei sau mai multe medii

```
165  ## testarea diferenței dintre 3 sau mai multe medii
166  from statsmodels.formula.api import ols
167  model = ols('area~stories_cat', data=houses_df_final).fit()
168  import statsmodels.api as sms
169  print(sms.stats.anova_lm(model, typ=2))
```

Figura 74. Input code

	sum_sq	df	F	PR(>F)
stories_cat	7.118300e+07	2.0	7.028451	0.001012
Residual	1.843266e+09	364.0	NaN	NaN

Figura 75. Output testarea diferentei dintre mediile categoriilor variabilei stories_cat in functie de variabila area

1. Formularea ipotezelor

- H0: $\mu1=\mu2=\mu3$ (nu există diferențe semnificative între suprafața medie a terenului casei în funcție de categoriile variabilei *store_cat*)
- H1: $\mu1\neq\mu2\neq\mu3$ (cel putin doua medii difera semnificativ intre ele. Există diferențe semnificative între suprafața medie a terenului casei în funcție de categoriile variabilei *store_cat*)

2. Testul folosit: F

3. Regula de decizie

- P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%
- $P_{value} \geq$ 0,05, nu se respinge H0, cu o probabilitate de 95%

  SAU
- $F_{calc} > F_t$, se respinge H0, cu un risc asumat de 5%
- $F_{calc} \leq F_t$, nu se respinge H0, cu o probabilitate de 95%

4. Decizie

P_{value} = 0,001 < α = 0,05, se respinge H0

5. Interpretare

Cu un risc asumat de 5%, putem spune ca există diferențe semnificative între suprafața medie a terenului casei în funcție de categoriile variabilei *store_cat*.

6. Analiza de regresie și corelație

În acest capitol se vor determina coeficienții de corelație, se vor testa și se vor realiza modele de regresie atât liniare cât și neliniare iar apoi se vor compara pentru a alege cel mai bun model.

6.1. Analiza de corelație

6.1.1. Matricea coeficienților de corelație

```
172  ################## 6. ANALIZA DE REGRESIE SI CORELATIE ##################
173  ########################################################################
174  ######  ANALIZA DE CORELATIE  ######
175  # Matrice coeficient de corelatie
176  import seaborn as sns
177  sns.heatmap(houses_df_num.corr(),square = True,annot = True, vmax=0.8)
```

Figura 76. Input code

Figura 77. Matricea coeficientilor de corelatie

Corelația dintre *price* și *area* este 0,53, ceea ce indică faptul că acestea sunt mediu corelate pozitiv. Un pret mai mare este dat de suprafață mai mare pe care o casă o poate avea. Această corelație medie pozitivă se regăsește și între *price* și *bathrooms*.

Corelația dintre *price* și *bedrooms* este 0,39, ceea ce indică faptul că acestea sunt corelate pozitiv scăzut. Această corelație scăzută pozitivă se regăsește și între *price* cu *parking*, *area* cu *bedrooms*, *area* cu *bathrooms*, *area* cu *parking*, *bedrooms* cu *bathrooms* și *bedrooms* cu *parking*.

6.1.2. Testarea coeficientul de corelație

```
178  # Testarea coeficientul de corelatie
179  from scipy.stats import pearsonr
180  print(pearsonr(houses_df_final.price,houses_df_final.area))
181  print(pearsonr(houses_df_final.price,houses_df_final.bedrooms))
182  print(pearsonr(houses_df_final.price,houses_df_final.bathrooms))
183  print(pearsonr(houses_df_final.price,houses_df_final.parking))
184  print(pearsonr(houses_df_final.area,houses_df_final.bedrooms))
185  print(pearsonr(houses_df_final.area,houses_df_final.bathrooms))
186  print(pearsonr(houses_df_final.area,houses_df_final.parking))
187  print(pearsonr(houses_df_final.bedrooms,houses_df_final.bathrooms))
188  print(pearsonr(houses_df_final.bedrooms,houses_df_final.parking))
189  print(pearsonr(houses_df_final.bathrooms,houses_df_final.parking))
```

Figura 78. Inpute code

```
In [99]: print(pearsonr(houses_df_final.price,houses_df_final.area))
    ...: print(pearsonr(houses_df_final.price,houses_df_final.bedrooms))
    ...: print(pearsonr(houses_df_final.price,houses_df_final.bathrooms))
    ...: print(pearsonr(houses_df_final.price,houses_df_final.parking))
    ...: print(pearsonr(houses_df_final.area,houses_df_final.bedrooms))
    ...: print(pearsonr(houses_df_final.area,houses_df_final.bathrooms))
    ...: print(pearsonr(houses_df_final.area,houses_df_final.parking))
    ...: print(pearsonr(houses_df_final.bedrooms,houses_df_final.bathrooms))
    ...: print(pearsonr(houses_df_final.bedrooms,houses_df_final.parking))
    ...: print(pearsonr(houses_df_final.bathrooms,houses_df_final.parking))
(0.5253148410993634, 2.0144881586137433e-27)
(0.3852746787449793, 1.9623864469155454e-14)
(0.5446164952682919, 9.891704174909204e-30)
(0.3571206651805369, 1.7637113461822948e-12)
(0.12051901286989986, 0.020923641553802667)
(0.2004271936392471, 0.00011071117985920571)
(0.29310622373393047, 1.052095884063617e-08)
(0.41170858948927336, 1.90280638057805e-16)
(0.13695824986181446, 0.008609503659899289)
(0.1991053810036968, 0.0001231284523494041)
```

Figura 79. Testare coeficient de corelație

1. **Formularea ipotezelor**
   - H0: $\rho=0$ (între variabile nu există o legătură semnificativă)
   - H1: $\rho\neq0$ (variabilele sunt corelate semnificativ)
2. **Testul folosit:** testul t Student
3. **Alegerea pragului de semnificație α**

$t_{teoretic} = t_{\alpha/2;n-2}$

4. Regula de decizie

- $P_{value} < 0,05$, se respinge H0, cu un risc asumat de 5%
- $P_{value} \geq 0,05$, nu se respinge H0, cu o probabilitate de 95%

SAU

- $t_{calc} > t_{teoretic}$, se respinge H0, cu un risc asumat de 5%
- $t_{calc} \leq t_{teoretic}$, nu se respinge H0, cu o probabilitate de 95%

5. Decizie

$P_{value\ price\&area} = 0,00... < \alpha = 0,05$, se respinge H0

$P_{value\ price\&bedrooms}$ = 0,00... < α = 0,05, se respinge H0

$P_{value\ price\&bathrooms}$ = 0,00... < α = 0,05, se respinge H0

$P_{value\ price\&parking}$ = 0,00... < α = 0,05, se respinge H0

$P_{value\ area\&bedrooms}$ = 0,02 < α = 0,05, se respinge H0

$P_{value\ area\&bathrooms}$ = 0,0001 < α = 0,05, se respinge H0

$P_{value\ area\&parking}$ = 0,00... < α = 0,05, se respinge H0

$P_{value\ bedrooms\&bathrooms}$ = 0,00... < α = 0,05, se respinge H0

$P_{value\ bedrooms\&parking}$ = 0,008 < α = 0,05, se respinge H0

$P_{value\ bathrooms\&parking}$ = 0,0001 < α = 0,05, se respinge H0

6. Interpretare

Cu un risc sumat de 5%, putem spune ca variabilele sunt corelate semnificativ, există legătură semnificativă între fiecare dintre cele 5 variabile analizate.

6.2. Analiza de regresie

6.2.1. Regresie liniară simplă

```
191  ######  ANALIZA DE REGRESIE  ######
192  ## regresie liniar simpla
193  import statsmodels.api as sm
194  Y = houses_df_final.price
195  X = houses_df_final.area
196  X = sm.add_constant(X)
197  model = sm.OLS(Y,X)
198  results = model.fit()
199  print(results.summary())
200  print('Parametrii',results.params)
201  print('R2',results.rsquared)
```

Figura 80. Input code

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.276
Model:                            OLS   Adj. R-squared:                  0.274
Method:                 Least Squares   F-statistic:                     139.1
Date:                Tue, 13 Dec 2022   Prob (F-statistic):           2.01e-27
Time:                        20:45:24   Log-Likelihood:                -5748.4
No. Observations:                 367   AIC:                         1.150e+04
Df Residuals:                     365   BIC:                         1.151e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       2.568e+06      2e+05     12.836      0.000    2.17e+06    2.96e+06
area         415.0601     35.191     11.795      0.000     345.858     484.262
==============================================================================
Omnibus:                       65.128   Durbin-Watson:                   0.540
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              131.737
Skew:                           0.944   Prob(JB):                     2.48e-29
Kurtosis:                       5.247   Cond. No.                     1.42e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.42e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Figura 81. Output summary

Dep. Variable: variabila dependenta aleasă este *price.*

Model și Method: Ordinary Least Square. Modelul încearcă să găsească o expresie liniară pentru setul de date care minimizează suma pătratelor reziduale.

Df Model: Din 2 variabile, doar o singură variabilă este independentă.

De asemenea, valoarea **testului F** pentru model este mai mic decât 0,05 de unde rezultă că modelul este corect specificat statistic.

Log-likelihood: Cu cât valoarea log-probabilității este mai mare, cu atât modelul se potrivește mai bine cu datele date.

AIC și BIC: Scopul este de a minimiza aceste valori pentru a obține un model mai bun.

P>|t|: Valoarea de 0,00 pentru *area* ne spune că există 0% șanse ca variabia *area* să nu aibă efect asupra prețului unei case (*price*).

Prob(Omnibus): reziduurile sunt perfect normale deoarece are valoarea 0.

Skew: Distribuția *area* este asimetrică la dreapta deoarece 0,94 > 0.

Kurtosis: variabile *area* este leptocurtică, valoarea aferentă boltirii 5,247 > 0.

```
Parametrii const    2.568298e+06
area     4.150601e+02
dtype: float64
```

Figura 82. Ecuatia estimată a parametrilor

Ecuația modelului de regresie: **Yi= β0+ βi*xi+ ε**

Ecuația modelului din analiza noastră: **price = 2.568.298+ 415 *(area)**

β0: Prețul unei case este 2.568.298$ atunci când suprafeței terenului (*area*) este egală cu 0.

β1: Prețul unei case crește cu 415$ la o creștere cu 1m^2 a suprafeței terenului.

```
R2 0.2759556822792495
```

Figura 83. Raportul de determinație

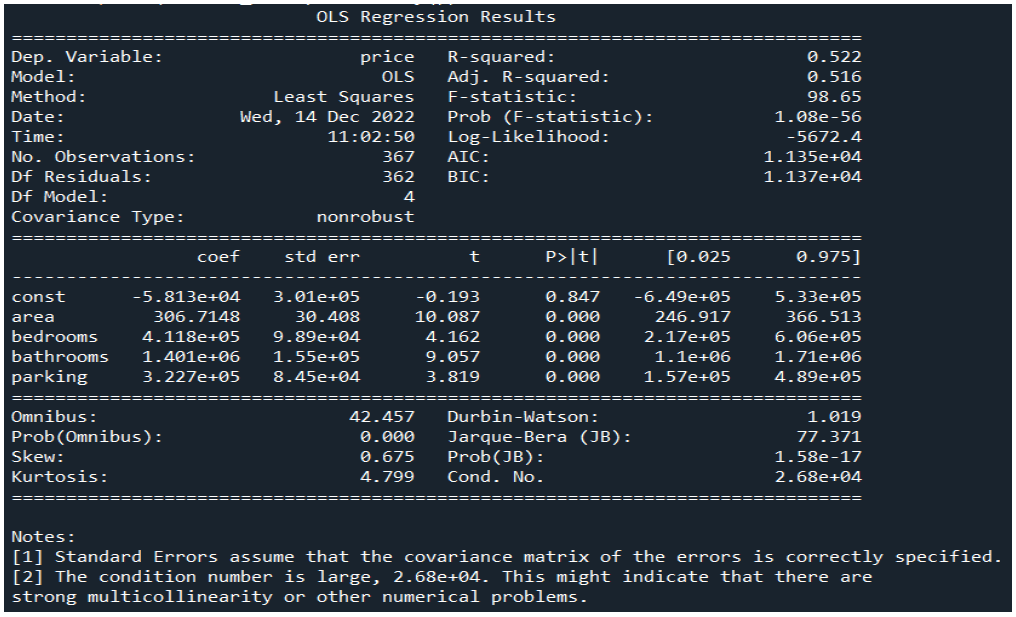
R^2 = 0,2759, acesta ne arată faptul că 27,59% din variația variabilei dependente *price* este explicată de variația variabilei independente *area*.

6.2.2. Regresie liniară multiplă

```
## regresie liniar multipla
from pandas import DataFrame
Y = houses_df_final.price
X_multiple=DataFrame({
    'area': houses_df_final.area,
    'bedrooms': houses_df_final.bedrooms,
    'bathrooms': houses_df_final.bathrooms,
    'parking': houses_df_final.parking})

X_multiple= sm.add_constant(X_multiple)
model_multiple=sm.OLS(Y, X_multiple)
results_multiple=model_multiple.fit()
print(results_multiple.summary())
print('Parametrii',results_multiple.params)
print('R2',results_multiple.rsquared)
```

Figura 84. Input code



OLS Regression Results

Dep. Variable:	price	R-squared:	0.522
Model:	OLS	Adj. R-squared:	0.516
Method:	Least Squares	F-statistic:	98.65
Date:	Wed, 14 Dec 2022	Prob (F-statistic):	1.08e-56
Time:	11:02:50	Log-Likelihood:	-5672.4
No. Observations:	367	AIC:	1.135e+04
Df Residuals:	362	BIC:	1.137e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-5.813e+04	3.01e+05	-0.193	0.847	-6.49e+05	5.33e+05
area	306.7148	30.408	10.087	0.000	246.917	366.513
bedrooms	4.118e+05	9.89e+04	4.162	0.000	2.17e+05	6.06e+05
bathrooms	1.401e+06	1.55e+05	9.057	0.000	1.1e+06	1.71e+06
parking	3.227e+05	8.45e+04	3.819	0.000	1.57e+05	4.89e+05

Omnibus:	42.457	Durbin-Watson:	1.019
Prob(Omnibus):	0.000	Jarque-Bera (JB):	77.371
Skew:	0.675	Prob(JB):	1.58e-17
Kurtosis:	4.799	Cond. No.	2.68e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.68e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Figura 85. Output summary

Dep. Variable: variabila dependenta aleasă este *price.*

Model și Method: Ordinary Least Square. Modelul încearcă să găsească o expresie liniară pentru setul de date care minimizează suma pătratelor reziduale.

Df Model: Din 5 variabile, 4 variabile sunt independente.

De asemenea, valoarea **testului F** pentru model este mai mică decât 0,05 de unde rezultă că modelul ste corect specificat statistic.

Log-likelihood: Cu cât valoarea log-probabilității este mai mare, cu atât modelul se potrivește mai bine cu datele date.

AIC și BIC: Scopul este de a minimiza aceste valori pentru a obține un model mai bun.

P>|t|: Valoarea de 0,00 pentru *area* și ne spune că există 0% șanse ca variabia *area* să nu aibă efect asupra prețului unei case (*price*). Același lucru se poate spune și despre variabila *bedrooms*, *bathrooms* și *parking*.

Prob(Omnibus): reziduurile sunt perfect normale deoarece are valoarea 0.

Skew: Distribuția *area* este asimetrică la dreapta deoarece 0,68 > 0.

Kurtosis: variabile *area* este leptocurtică, valoarea aferentă boltirii 0,80 > 0.

```
Parametrii const       -5.813302e+04
area          3.067148e+02
bedrooms      4.117873e+05
bathrooms     1.401223e+06
parking       3.227338e+05
dtype: float64
```

Figura 86. Ecuatia estimată a parametrilor

Ecuația modelului de regresie multiplă : **Y= β0 + β1X1 + β2X2 + β3X3 + β4X4 + εi**

Ecuația modelului din analiza noastră: **price = -58.133 + 307*area + 41.787*bedrooms + 1.401.223*bathrooms + 322.733*parking**

β0: Valoarea așteptată a prețului unei case va fi mai mică decât 0 atunci când toate variabilele independente/predictoare sunt egale cu 0.

β1: Prețul unei case crește cu 307$ la o creștere cu 1 m^2 a suprafeței terenului (*area*), în condițiile în care influența variabilelor *bedrooms, bathrooms* și *parking* rămân constane.

β2: Prețul unei case crește cu 41787$ la o creștere cu 1 cameră de dormitor (*bedrooms*), în condițiile în care influența variabilelor *area, bathrooms* și *parking* rămân constane.

β3: Prețul unei case crește cu 1401223$ la o creștere cu 1 baie (*bethrooms*), în condițiile în care influența variabilelor *area, bedrooms* și *parking* rămân constane.

β4: Prețul unei case crește cu 322733$ la o creștere cu 1 loc de parcare (*parking*), în condițiile în care influența variabilelor *area, bedrooms* și *bathrooms* rămân constane.

```
R2 0.5215332530427743
```

Figura 87. Raportul de determinație

$\mathbf{R^2}$ = 0,5215, acesta ne arată faptul că 52,15% din variația variabilei dependente *price* este explicată de variația variabilelor independente *area, bedrooms, bathrooms* și *parking*.

6.2.3. Regresie neliniară

Voi realiza un model parabolic între variabila *price* și *area*.

```
219  ## regresie neliniara
220  X_nel = DataFrame({'area' : houses_df_final.area, 'area^2' : houses_df_final.area**2 })
221  X_nel = sm.add_constant(X_nel)
222  Y = houses_df_final.price
223  model_nel = sm.OLS(Y, X_nel)
224  results_nel = model_nel.fit()
225  print(results_nel.summary())
226  print('Parametrii',results_nel.params)
227  print('R2',results_nel.rsquared)
```

Figura 88. Input code

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.306
Model:                            OLS   Adj. R-squared:                  0.302
Method:                 Least Squares   F-statistic:                     80.32
Date:                Wed, 14 Dec 2022   Prob (F-statistic):           1.28e-29
Time:                        11:36:48   Log-Likelihood:                -5740.6
No. Observations:                 367   AIC:                         1.149e+04
Df Residuals:                     364   BIC:                         1.150e+04
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.119e+06   4.13e+05      2.708      0.007    3.07e+05    1.93e+06
area         916.7787    130.628      7.018      0.000     659.898    1173.659
area^2        -0.0360      0.009     -3.982      0.000      -0.054      -0.018
==============================================================================
Omnibus:                       59.966   Durbin-Watson:                   0.620
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              116.732
Skew:                           0.890   Prob(JB):                     4.49e-26
Kurtosis:                       5.113   Cond. No.                     2.42e+08
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.42e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Figura 89. Output summary

Din output se poate observa că atât constanta modelului cât și coeficientul variabilei independente sunt semnificativi statistic, ambii având probabilități mai mici decât 0,05. De asemenea, valoarea testului F este mai mică decât 0,05 de unde rezultă că modelul este corect specificat statistic.

```
Parametrii const     1.119489e+06
area       9.167787e+02
area^2    -3.599298e-02
dtype: float64
```

Figura 90. Ecuatia estimată a parametrilor

Ecuatia generala a modelului: $\mathbf{Y= \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \varepsilon_i}$

La nivelul esantionului: $\mathbf{Y = b_0 + b_1 X_1 + b_2 X_1^2}$

Ecuația modelului din analiza noastră: **price = 1119489 + 917*area - (0,04*area)**

```
R2 0.3061819140824452
```

Figura 91. Raportul de determinație

$\mathbf{R^2}$ = 0,3061, astfel că putem spune că modelul explică 30,61% din variația variabilei dependente *price*.

6.2.4. Testare ipoteze model de regresie liniară simplă

```
229  ######  TESTARE IPOTEZE  ######
230  ##--> Regresie liniar simpla
231  # erori
232  print('Parameters:', results.params)
233  print('R2:', results.rsquared)
234  print('Predicted values:', results.predict())
235  print('Erori de modelare:', results.resid)
236  # Salvarea reziduurilor
237  erori_rls = results.resid
```

Figura 92. Input code

A. Testarea ipotezei privind media erorilor este nula M(εi)=0

```
239  # Testarea ipotezei privind media erorilor este nula
240  import scipy.stats as stats
241  print(stats.ttest_1samp(erori_rls, 0))
```

Figura 93. Input code

```
Ttest_1sampResult(statistic=3.035170400961226e-15, pvalue=0.9999999999999976)
```

Figura 94. Output

1. Formularea ipotezelor

- H0: M (εi) = 0 (media erorilor este 0)
- H1: M (εi) ≠ 0 (media erorilor este diferită de 0)

2. Testul folosit: testul t

3. Regula de decizie

- P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%
- P_{value} ≥ 0,05, nu se respinge H0, cu o probabilitate de 95%

  SAU
- $t_{calc} > t_{teoretic}$, se respinge H0, cu un risc asumat de 5%
- $t_{calc} \leq t_{teoretic}$, nu se respinge H0, cu o probabilitate de 95%

4. Decizie

P_{value} = 0.99 > α = 0,05, nu se respinge H0

5. Interpretare

Cu o probabilitate de 95%, se acceptă ipoteza conform căreia media erorilor este egală cu zero. În concluzie, nu se modifică proprietățile estimatorilor parametrilor modelului de regresie.

B. Testarea ipotezei de normalitate a erorilor

```
243  # Testarea ipotezei de normalitate a erorilor
244  from scipy.stats import normaltest
245  print(normaltest(erori_rls))
```

Figura 95. Input code

```
NormaltestResult(statistic=65.12763309701268, pvalue=7.206330287938433e-15)
```

Figura 96. Output

1. Formularea ipotezelor

- H0: Erorile urmează o lege normală de distribuție
- H1: Erorile nu urmează o lege normală de distribuție

2. Testul folosit: JB

3. Regula de decizie

- $P_{value} < 0,05$, se respinge H0, cu un risc asumat de 5%
- $P_{value} \geq 0,05$, nu se respinge H0, cu o probabilitate de 95%

  SAU
- $JB_{calc} > X^2$, se respinge H0, cu un risc asumat de 5%
- $JB_{calc} \leq X^2$, nu se respinge H0, cu o probabilitate de 95%

4. Decizie

$P_{value} = 0.00 < \alpha = 0,05$, se respinge H0

5. Interpretare

Deoarece probabilitatea asociata testului Jarque Bera este 0,00 atunci se va lua decizia de a se respinge ipoteza nulă, astfel încât rezultă că erorile nu urmează o lege de distribuție normală.

C. Testarea ipotezei de homoscedasticitate a erorilor

```
247  # Testarea ipotezei de homoscedasticitate a erorilor
248  import statsmodels.stats.api as sms
249  test_GQ=sms.het_goldfeldquandt(erori_rls, results.model.exog)
250  print(test_GQ)
```

Figura 97. Input code

```
(0.16658835725557936, 0.9999999999999999, 'increasing')
```

Figura 98. Output

1. **Formularea ipotezelor**
   - H0: Erorile sunt homoscedastice (V(εi)= σ2)
   - H1: Erorile nu sunt homoscedastice (sunt heteroscedastice)
2. **Testul folosit:** Goldfeld-Quandt
3. **Regula de decizie**
   - P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%
   - P_{value} ≥ 0,05, nu se respinge H0, cu o probabilitate de 95%
4. **Decizie**

P_{value} = 0.99 > α = 0,05, nu se respinge H0

5. **Interpretare**

Întrucât este egal cu 0,99 care este mai mare decât riscul asumat de α = 0,05 nu se respinge ipoteza nulă cu un risc asumat de 5%. Așadar, erorile sunt homoscedastice. astfel estimatorii parametrilor de regresie nu își pierd eficiența.

D. Testarea autocorelarii erorilor

```
252  # Testarea autocorelarii erorilor
253  import statsmodels.tsa.api as smt
254  acf=smt.graphics.plot_acf(erori_rls, lags=10, alpha=0.05)
255  acf.show()
```

Figura 99. Input code

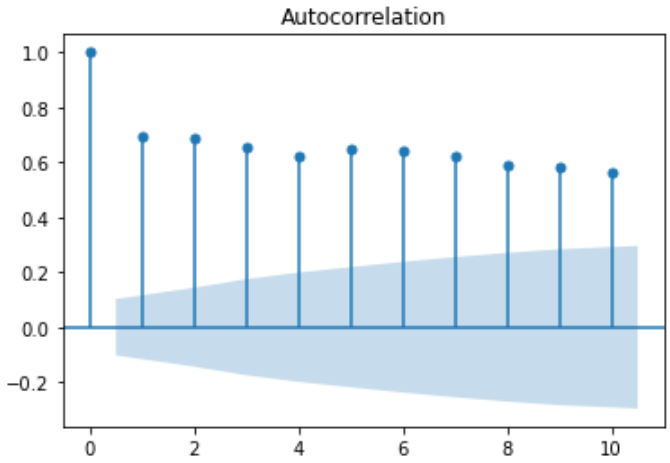


Figura 100. Output

Graficul de autocorelare arată că autocorelațiile eșantionului sunt puternic pozitive și se degradează foarte lent.

6.2.5. Testare ipoteze model de regresie liniară multiplă

```
257    ##--> Regresie liniar multipla
258    # erori
259    print('Parameters:', results.params)
260    print('R2:', results.rsquared)
261    print('Predicted values:', results.predict())
262    print('Erori de modelare:', results.resid)
263    # Salvarea reziduriilor
264    erori_rlm = results_multiple.resid
```

Figura 101. Input code

A. Testarea ipotezei privind media erorilor este nula M(εi)=0

```
266    # Testarea ipotezei privind media erorilor este nula
267    import scipy.stats as stats
268    print(stats.ttest_1samp(erori_rlm, 0))
```

Figura 102. Input code

```
Ttest_1sampResult(statistic=7.84077637831514e-14, pvalue=0.9999999999999375)
```

Figura 103. Output

1. **Formularea ipotezelor**
   - H0: M (εi) = 0 (media erorilor este 0)
   - H1: M (εi) ≠ 0 (media erorilor este diferită de 0)
2. **Testul folosit:** testul t

3. Regula de decizie

- P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%
- P_{value} ≥ 0,05, nu se respinge H0, cu o probabilitate de 95%

SAU

- $t_{calc} > t_{teoretic}$, se respinge H0, cu un risc asumat de 5%
- $t_{calc} \leq t_{teoretic}$, nu se respinge H0, cu o probabilitate de 95%

4. Decizie

P_{value} = 0,99 > α = 0,05, nu se respinge H0

5. Interpretare

Cu o probabilitate de 95%, se acceptă ipoteza conform căreia media erorilor este egală cu zero. În concluzie, nu se modifică proprietățile estimatorilor parametrilor modelului de regresie

B. Testarea ipotezei de normalitate a erorilor

```
270  # Testarea ipotezei de normalitate a erorilor
271  from scipy.stats import normaltest
272  print(normaltest(erori_rlm))
```

Figura 104. Input code

```
NormaltestResult(statistic=42.457120610414975, pvalue=6.033278883525807e-10)
```

Figura 105. Output

1. Formularea ipotezelor

- H0: Erorile urmează o lege normală de distribuție
- H1: Erorile nu urmează o lege normală de distribuție

2. Testul folosit: JB

3. Regula de decizie

- $P_{value} < 0,05$, se respinge H0, cu un risc asumat de 5%
- $P_{value} \geq 0,05$, nu se respinge H0, cu o probabilitate de 95%

SAU

- $JB_{calc} > X^2$, se respinge H0, cu un risc asumat de 5%
- $JB_{calc} \leq X^2$, nu se respinge H0, cu o probabilitate de 95%

4. Decizie

$P_{value} = 0.00 < \alpha = 0,05$, se respinge H0

5. Interpretare

Deoarece probabilitatea asociata testului Jarque Bera este 0,00 atunci se va lua decizia de a se respinge ipoteza nulă, astfel încât rezultă că erorile nu urmează o lege de distribuție normală.

C. Testarea ipotezei de homoscedasticitate a erorilor

```
274  # Testarea ipotezei de homoscedasticitate a erorilor
275  import statsmodels.stats.api as sms
276  test_GQ=sms.het_goldfeldquandt(erori_rlm, results_multiple.model.exog)
277  print(test_GQ)
```

Figura 106. Input code

```
(0.21037851706624688, 0.9999999999999999, 'increasing')
```

Figura 107. Output

1. **Formularea ipotezelor**
   - H0: Erorile sunt homoscedastice (V(εi)= σ2)
   - H1: Erorile nu sunt homoscedastice (sunt heteroscedastice)
2. **Testul folosit:** testul t
3. **Regula de decizie**
   - P_{value} < 0,05, se respinge H0, cu un risc asumat de 5%
   - P_{value} ≥ 0,05, nu se respinge H0, cu o probabilitate de 95%

     SAU
   - $t_{calc} > t_{teoretic}$, se respinge H0, cu un risc asumat de 5%
   - $t_{calc} \leq t_{teoretic}$, nu se respinge H0, cu o probabilitate de 95%
4. **Decizie**

P_{value} = 0.99 > α = 0,05, nu se respinge H0

5. Interpretare

Întrucât este egal cu 0,99 care este mai mare decât riscul asumat de α = 0,05 nu se respinge ipoteza nulă cu un risc asumat de 5%. Așadar, erorile sunt homoscedastice. Astfel estimatorii parametrilor de regresie nu își pierd eficiența.

D. Testarea autocorelarii erorilor

```
279  # Testarea autocorelarii erorilor
280  import statsmodels.tsa.api as smt
281  acf=smt.graphics.plot_acf(erori_rlm, lags=10, alpha=0.05)
282  acf.show()
```

Figura 108. Input code

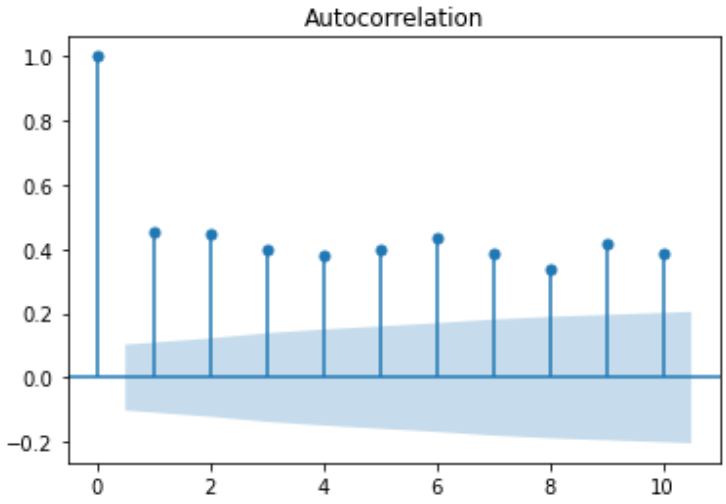


Figura 109. Output

. Graficul de autocorelare arată că autocorelațiile eșantionului sunt pozitive și se degradează lent.

6.3.Compararea modelelor de regresie și alegerea celui mai potrivit model

Pentru compararea celor 2 modele de regresie s-a realizat urmatorul tabel:

	Regresie liniar simplă	Regresie liniar multiplă
R^2	0.2759	0,5215
P_{value}	0,00	0,00

Se poate observa că modelul care explică cel mai mult variația variabilei dependente *price* este modelul multiplu, care prezintă un R^2 de 52,15%, în comparație cu modelul simplu care explică mai puțin din variația variabilei *price*.

7. Concluzii

Pe parcursul proiectului, am reușit să îndeplinesc obiectivele propuse la introducerea lucrării cu ajutorul analizelor statistice și a modelelor de regresie.

Observațiile analizate sunt alcătuite în cea mai mare parte de case fără subsol și aer condiționat și care au două magazii. Prețul mediu al unei case ajunge la 4.729.482$.

Prețul unei case este 2.568.298$ atunci când influența suprafeței terenului (*area*) este egală cu 0. În schimb, prețul acestora crește cu 415$ la o creștere cu $1m^2$ a suprafeței terenului.

Din matricea corelațiilor s-a observat existența unei legături medii și directe dintre preț și suprafața terenului unde este construită casa. Aceasta este egală cu 0,53. Un pret mai mare este dat de suprafață mai mare a unei case. Această corelație medie pozitivă se regăsește și între preț și numărul de băi pe care o casă o poate avea.

S-a testat intervalul de încredere al variabilelor și astfel am aflat că, cu o probabilitate de 95%, se poate afirma că intervalul de încredere al variabilei *price* este acoperit de valorile [4543989,26 , 4914974,17]$, intervalul de încredere al variabilei *area* este [4972,15 , 5441,68] m^2, intervalul de încredere al variabilei *bedrooms* este [2,84 , 2,99] de camere, intervalul de încredere al variabilei *bathrooms* este [1,21 , 1,31] de camere și intervalul de încredere al variabilei *parking* este [0,61 , 0,78] locuri de parcare.

Cu ajutorul testului ANOVA am putut observa că există diferențe semnificative între suprafața medie a terenului unei casei în funcție de categoriile variabilei *store_cat*.

În concluzie, se poate afirma că modelul liniar multiplu este considerat un bun model deoarece explica 52,15% din variația variabilei dependente *price* cu ajutorul variabilelor independente *area*, *bedrooms*, *bathrooms*, *parking* și că toți aceștia sunt factori care determină prețului a unei case.