PavelIoanCristian 311CAb

#Task_1
am citit informatiile din fisier (my_file = pandas.read_csv('train.csv'))
functia shape returneaza nr de linii si coloane sub forma de tuple
nr_col = my_file.shape[0] #coloane
nr_lin = my_file.shape[1] #linii
metoda .dtypes returneaza un series cu numele coloanele si data type ul din fiecare coloana
metoda .isnull() creeaza un dataframe cu true / false pentru valori lipsa
metoda .sum() transforma df ul intr-un series si aduna acele valori de true
metoda .duplicated().sum() un series cu numele col si True / False, unde valorile de True vor fi insumate

#Task_2
am accesat coloana cu supravietuitori cu my_file['Survived'] si folosit functia .mean() am facut media
am determinat apoi procentul inmultind cu 100; nonsupravietuitorii vor fii 100 - supravietuitori
pentru Pcclass si Sex am caculat procentajul de clasificare;
metoda .value_counts() calculeaza de cate ori apare fiecare valoare din coloana respectiva
normalize='True' transforma numarul de valori intr-o proportie
am creat apoi o figura 'fig' cu 3 subploturi, axs
pentru primul grafic:
['Supravietuitori', 'nonsupravietuitori'] - labelurile
[supravietuitori_percentage, nesupravietuitori_percentage] - valorile de reprezentat prin bars
color=['blue', 'orange'] - culorile pentru fiecare bar
salvam graficul ca imagine fig.savefig('grafic_task_2.png');

#Task_3
metoda .select_dtypes selecteaza din data frame coloanele cu tipurile de date precizate in include
pentru fiecare coloana selectata, se creeaza un grafic care reprezinta valoarea in functie de frecventa
ei in aparitia in coloana

#Task_4
my_file.isnull().any() seteaza cu True coloanele care au elemente lipsa
pasand valoarea ca argument la my_file.columns, selectam doar acele coloane din lista de coloane
metoda .tolist() transforma lista pandas intr-o lista python
pentru fiecare coloana din cele cu valori lipsa se calculeaza procentul supravietuitorilor si al nonsupravietuitorilor, la fel ca la taskul 2; de asemenea, se calculeaza si procentul valorilor lipsa,

#Task_5
pentru a reprezenta histograma trebuie sa definim labelurile cu range urile de varsta, cat si bin urile

label urile vor fi stringurile prezentate in cerinta; pentru binuri, avem nevoie de un capat superior drept,
reprezentat de limita maxima de varsta atinsa in data frame; o calculam prin metoda .max()
apoi folosim metoda .cut() pentru a imparti coloana Age in bin uri, cu labeluri din labels
numaram apoi valorile unice (persoanele) prin value_counts()
desenam graficul folosind bars, unde elemente de pe axa X sunt etichetele din dataframe (indexurile),
iar pe Y avem valorile efective

#Task_6
persoane_pe_categorie = my_file.groupby(['categorie-varsta', 'Sex'], observed=False)['Survived'].mean() * 100
se face gruparea persoanelor in functie de categoria de varsta si separarea dupa gen
observed=false nu permite adaugarea unor categorii care nu sunt prezente in setul de date
['Survived'].mean() * 100 acceseaza coloana survived din data frame si calculeaza procentul
se deseneaza apoi graficul, asemanator ca la task 5; plot este o metoda pandas folosita pentru grafice;

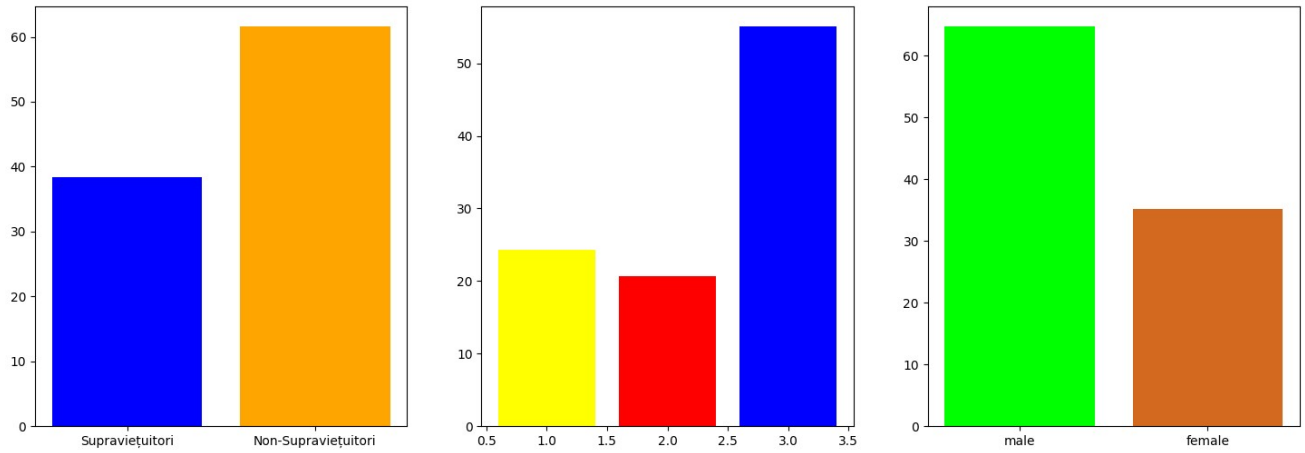
#Task_7
my_file[my_file['Age'] < 18] selecteaza doar randurile cu persoanele cu varsta < 18
procentajul copiilor se calculeaza impartind numarul de copii la numarul total de persoane si inmultind cu 100
procentul de supravietuire se calculeaza ca la subpunctul anterior
graficul se deseneaza ca la taskul 2;

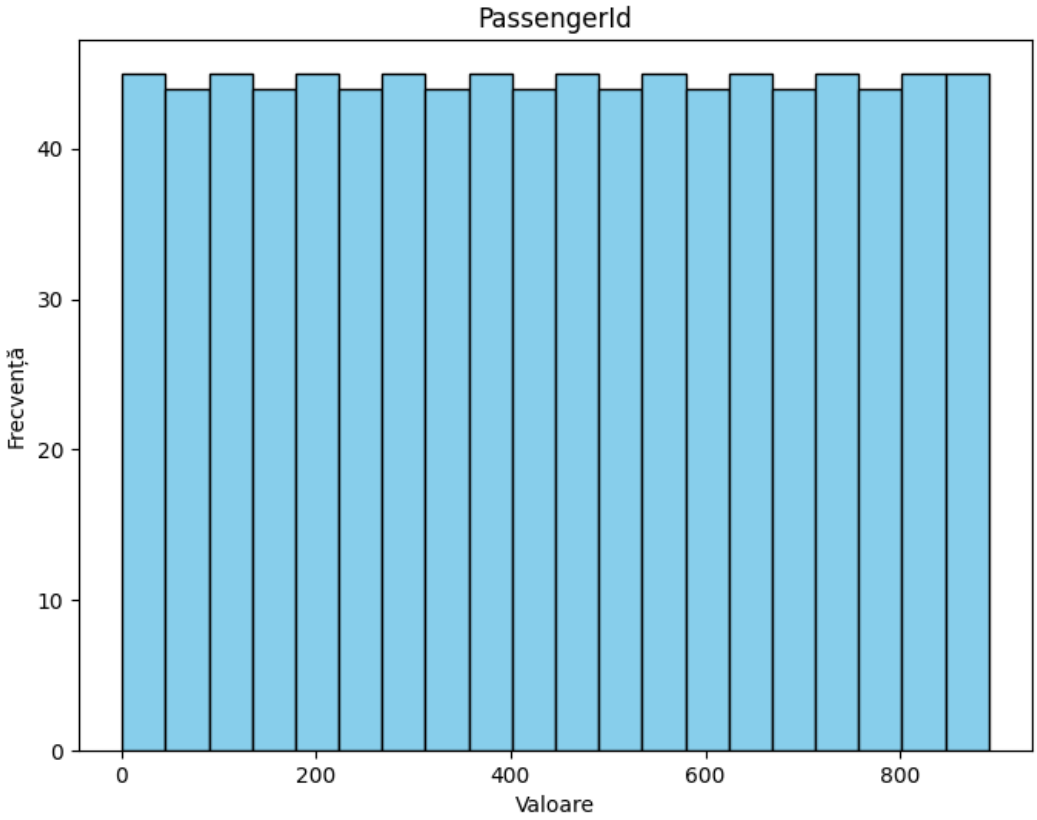
#Task_8
metoda .groupby(['Pclass', 'Survived']) grupeaza persoanele care au aceeasi clasa si stare de supravietuire
metoda .transform aplica o functie asupra fiecarei valori din coloanele selectate
pentru metoda transform avem nevoie de o functie, de aceea cream o functie lambda
metoda .filna(x.mean()) inlocuieste valorilea lipsa (care sunt setate ca NaN -> NotaNumber) cu media valorilor (x.mean())

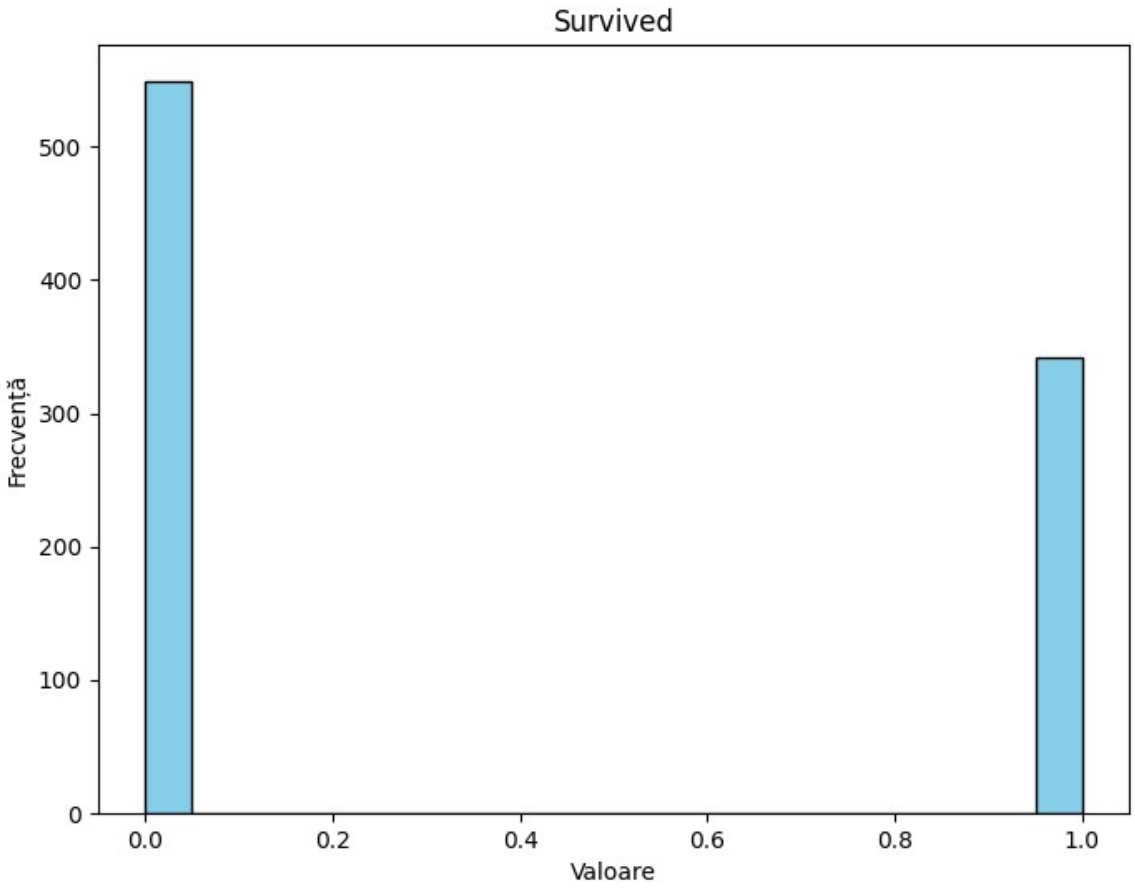
#Task_9
nume_noblime = my_file['Name'].str.split(',') #desparte textul in functie de virgule
nume_noblime = nume_noblime.str[1] #extrage partea care contine numele si prenumele
nume_noblime = nume_noblime.str.split('.') #desparte textul in functie de .
nume_noblime = nume_noblime.str[0].str.strip()
#str[0] este titlu de noblime, iar str.strip() sterge eventualele spatii albe de la inceput si sfarsit
se creeaza apoi coloana titlu_de_noblime
folosind metoda .groupby se grupeaza data frame ul dupa coloanele title_de_noblime si Sex
graficul se creeaza folosind metoda .plot() din pandas, ca la task 6

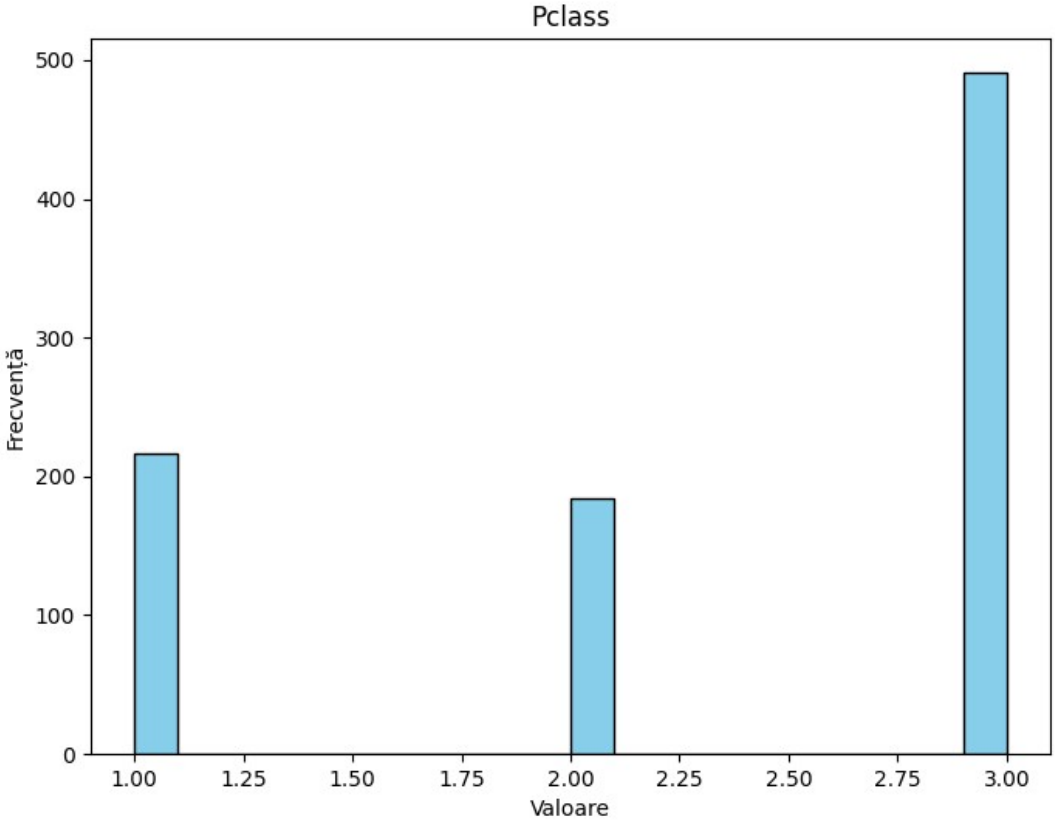
#Task_10
folosind catplot din seaborn se creeaza histograma;
data=my_file.head(100) specifica ca lucram cu primele 100 de randuri

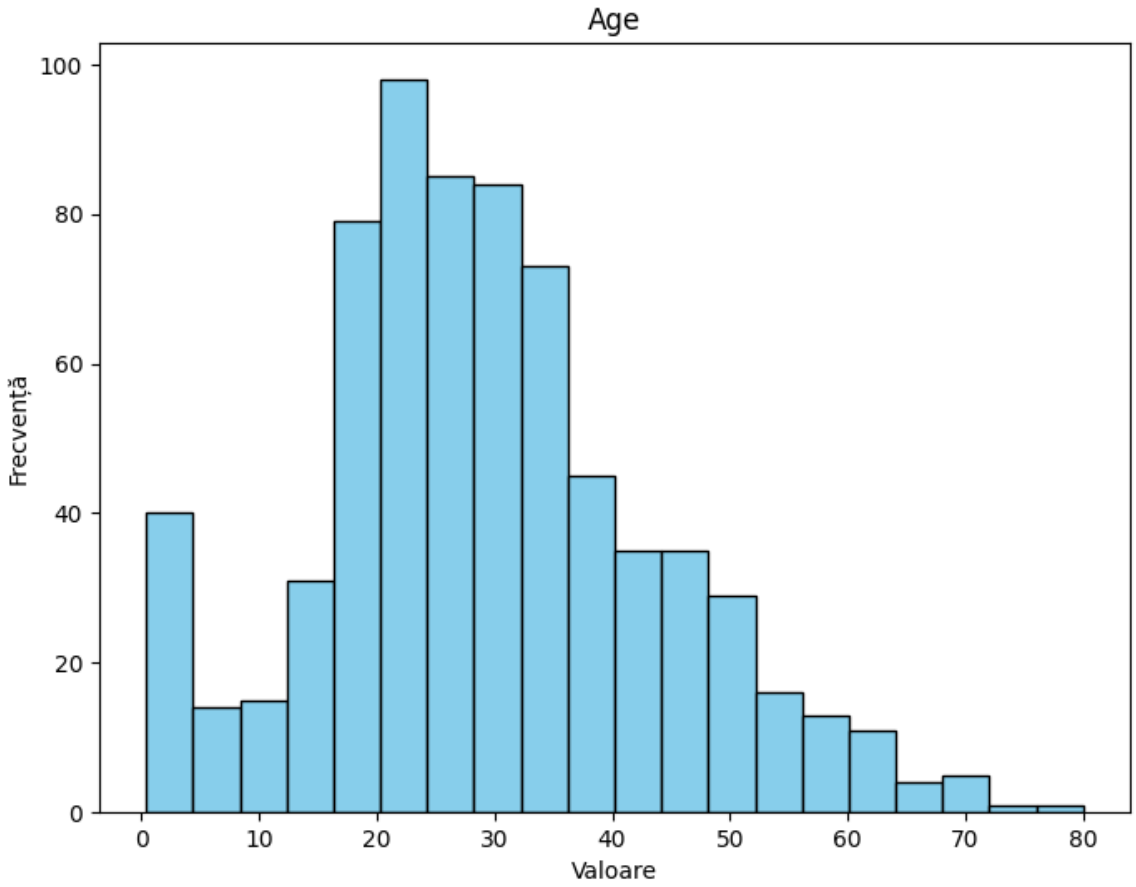
pclass e coloana plasata pe axa x a graficului
fare este coloana plasat pe axa y a graficului
survived este coloana folosita pentru a colora punctele
strip specifica ca vrem sa cream un grafic de puncte
jitter adauga o mica distantare intre puncte pentru a le face vizibile in caz ca se suprapun

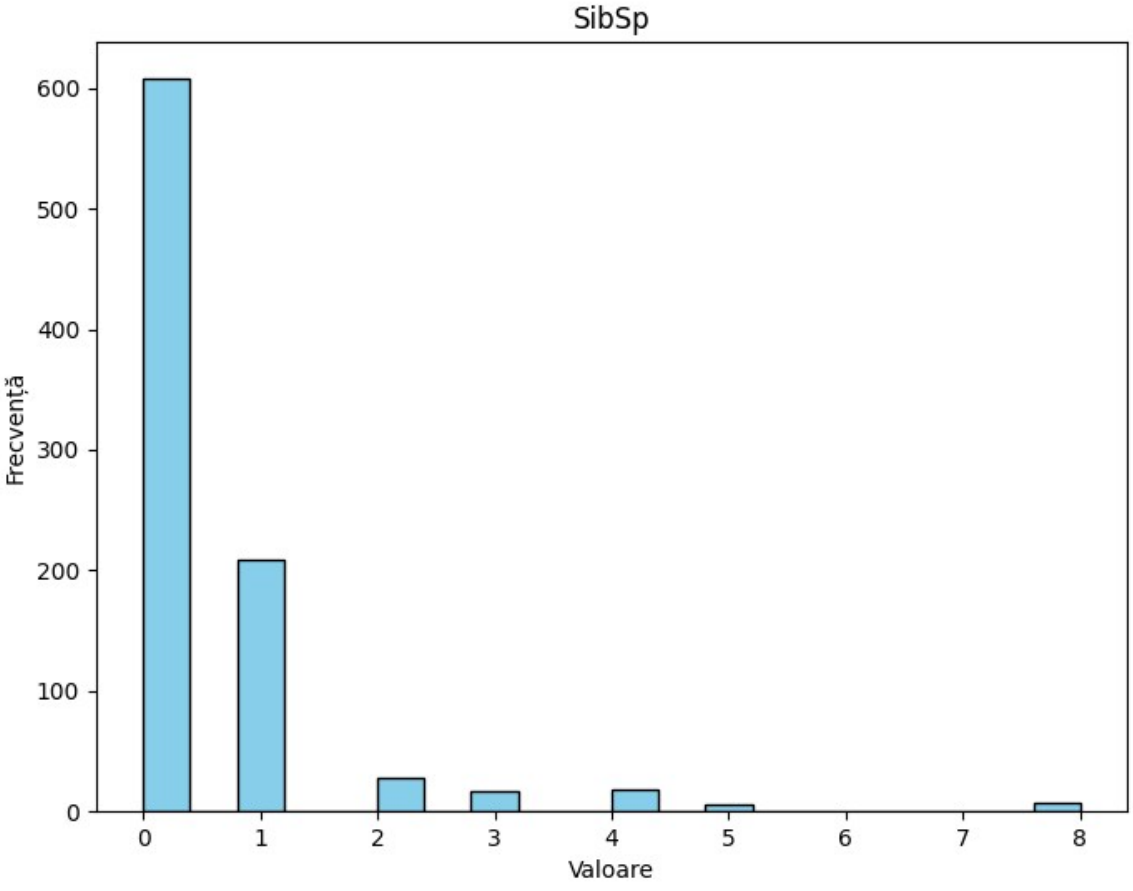


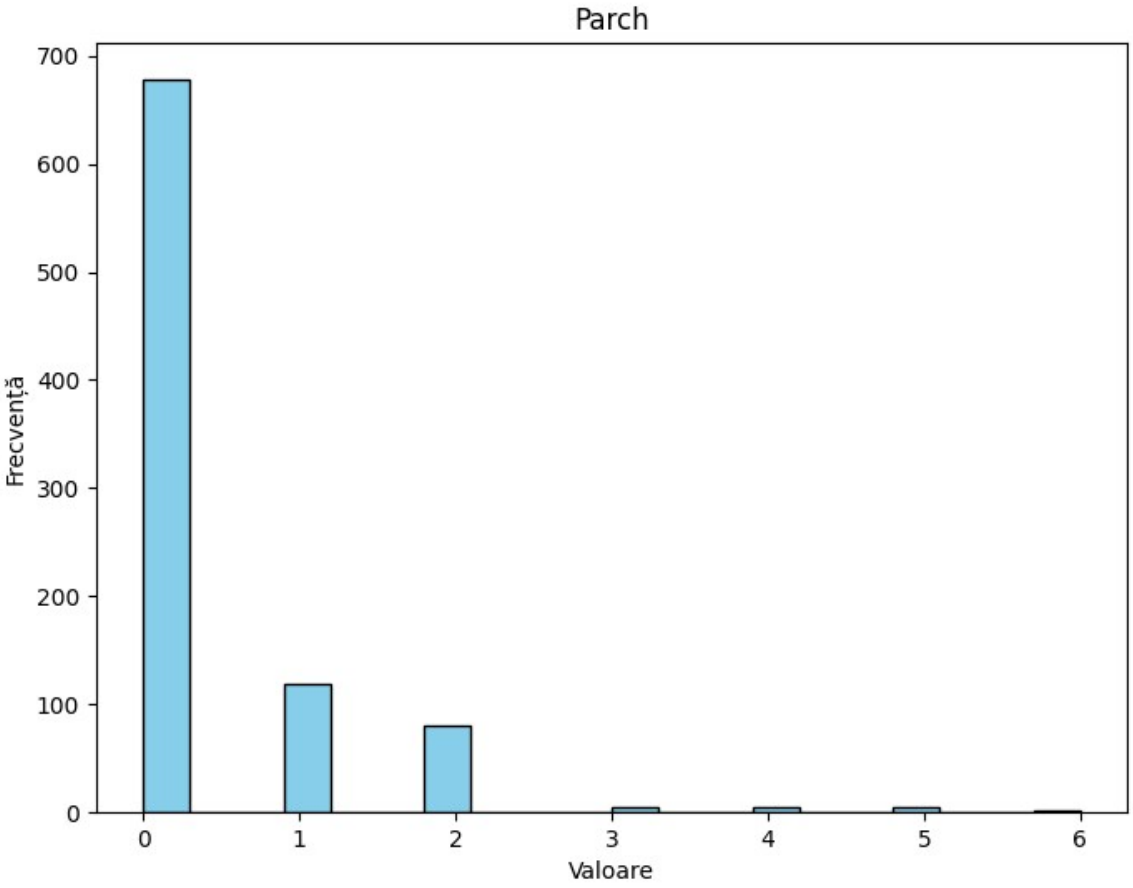


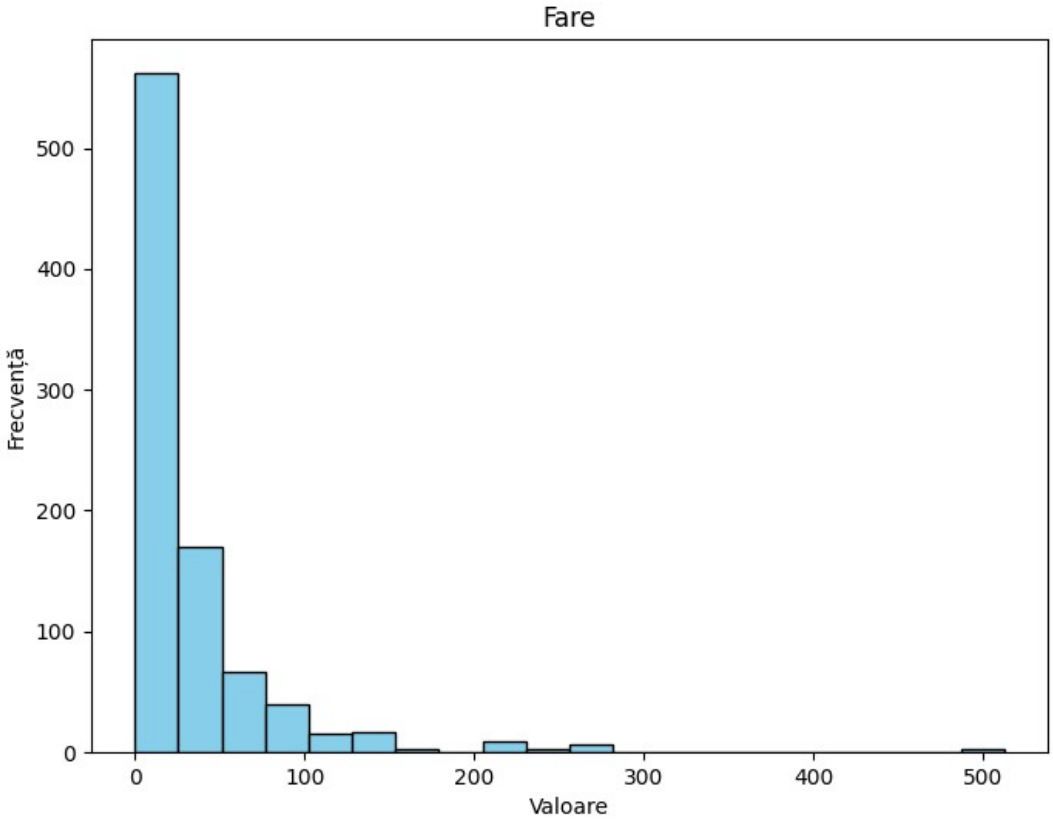

Survived
Frecvență
Valoare
0
100
200
300
400
500
0.0
0.2
0.4
0.6
0.8
1.0

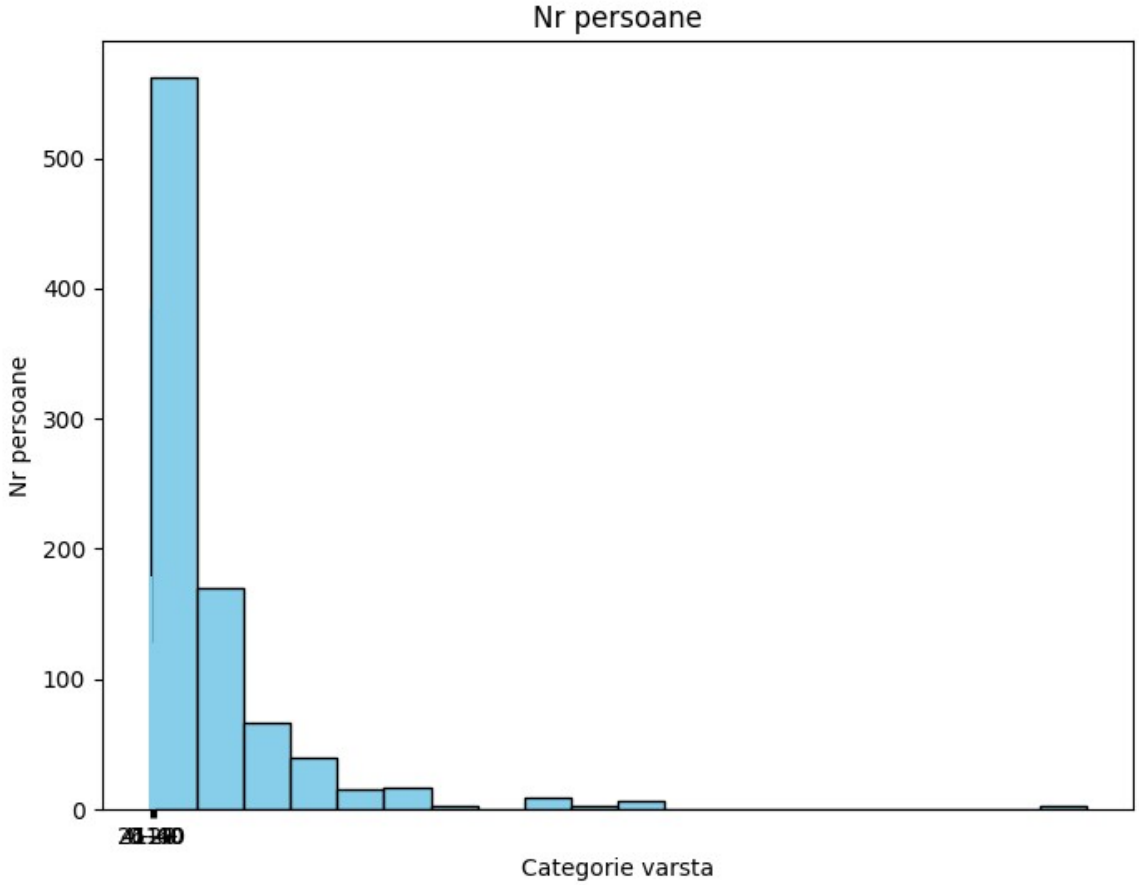

Pclass
Frecvență
Valoare
0
100
200
300
400
500
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00

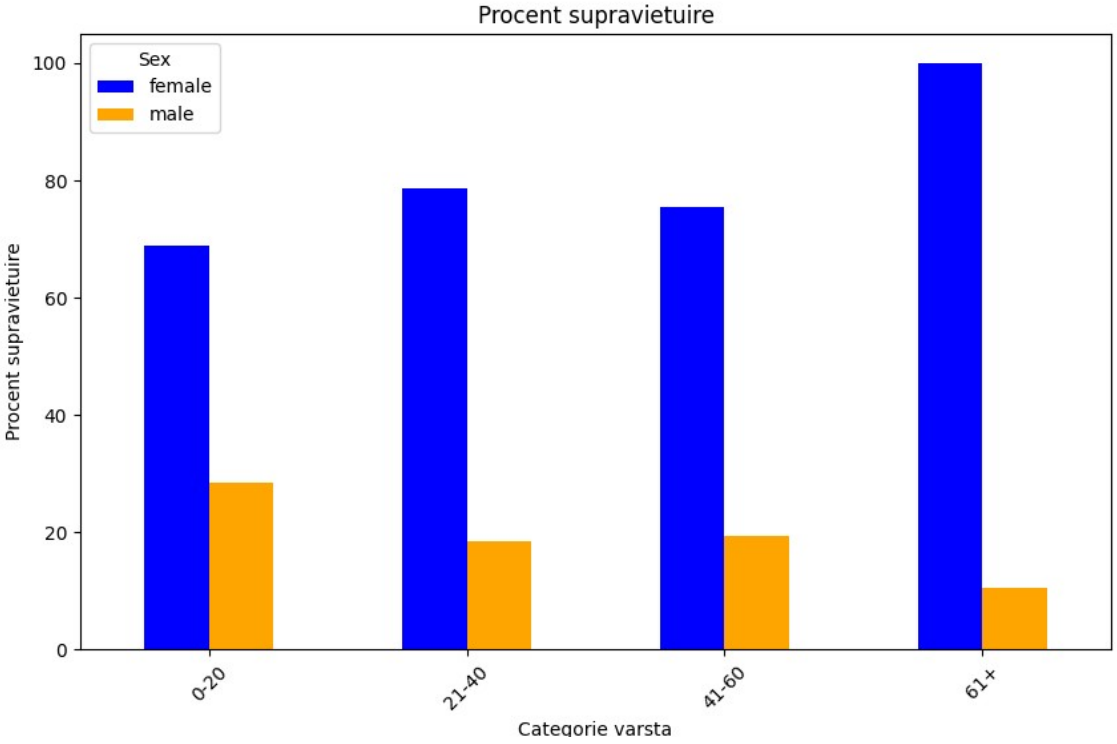

Age
Frecvență
Valoare
0
20
40
60
80
100
0
10
20
30
40
50
60
70
80

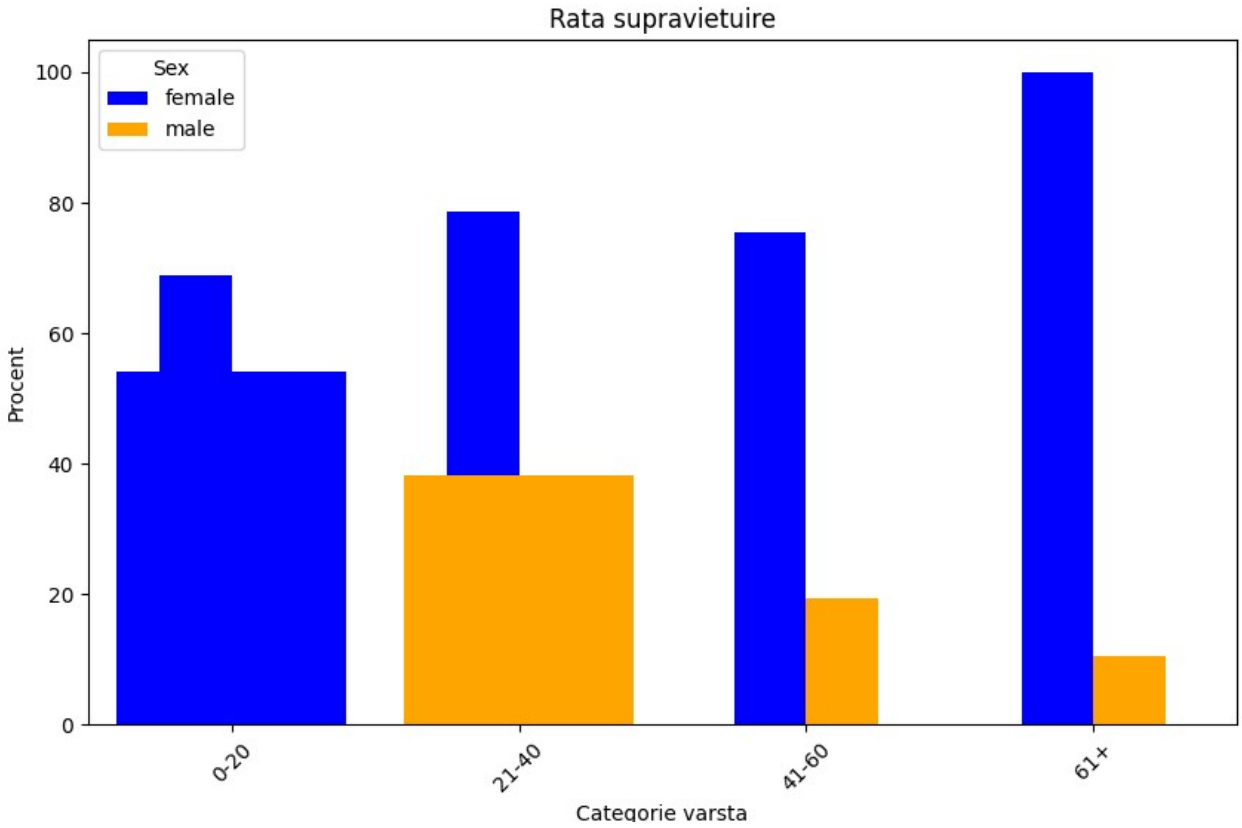

SibSp
Frecvență
Valoare
0
100
200
300
400
500
600
0
1
2
3
4
5
6
7
8

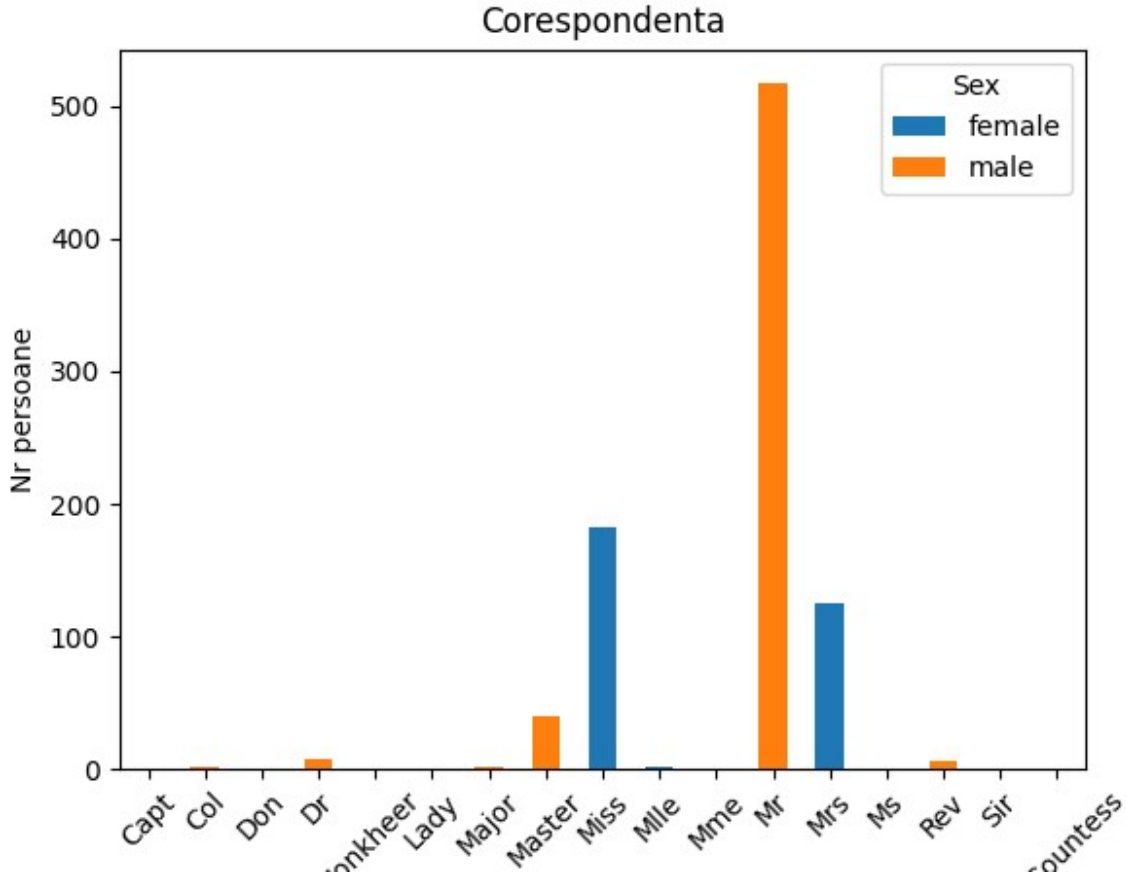

Parch
Frecvență
Valoare
0
100
200
300
400
500
600
700
0
1
2
3
4
5
6

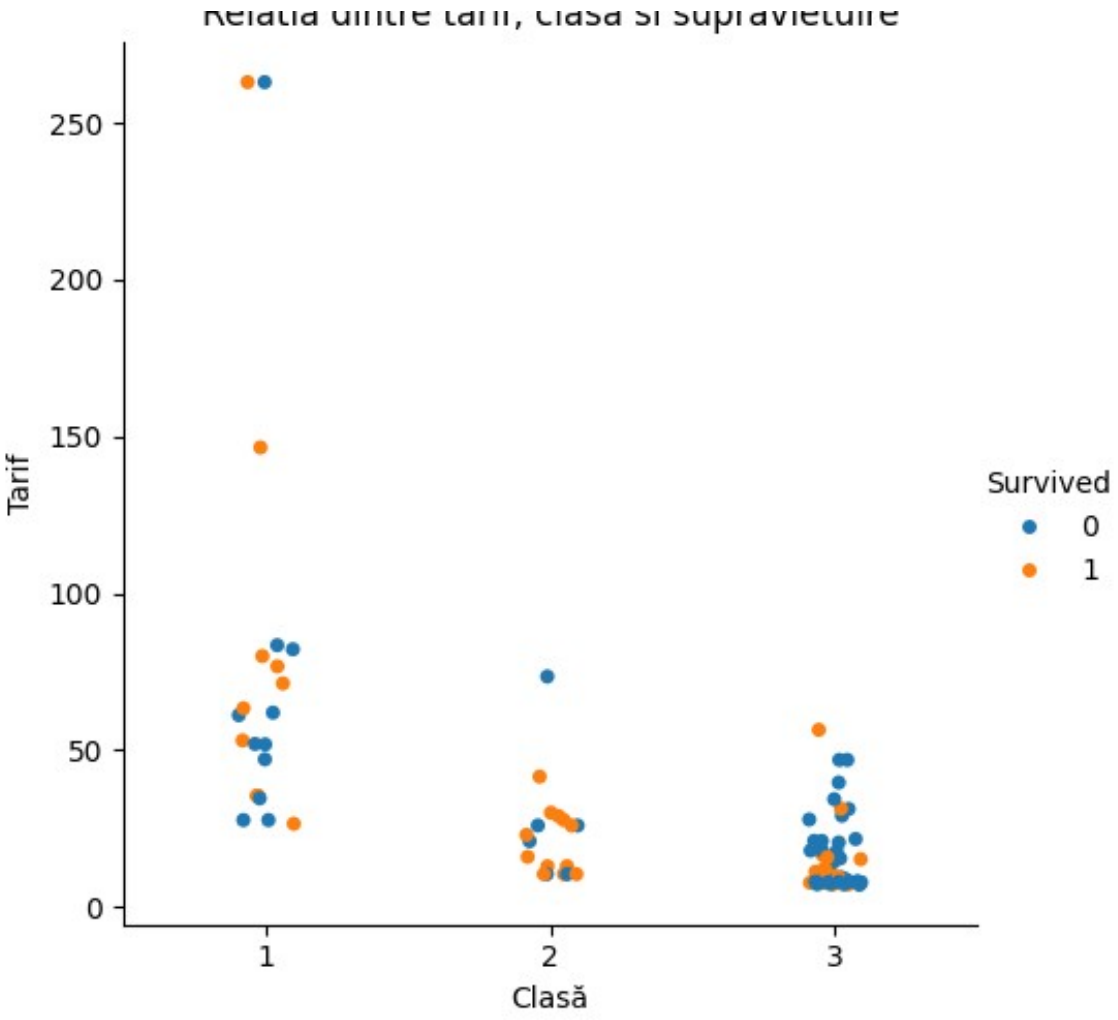

Fare
Frecvență
Valoare
0
100
200
300
400
500
0
100
200
300
400
500


Nr persoane
Nr persoane
500
400
300
200
100
0
Categorie varsta


Procent supravietuire
Sex
female
male
Procent supravietuire
100
80
60
40
20
0
0-20
21-40
41-60
61+
Categorie varsta


Rata supravietuire
Sex
female
male
Procent
0
20
40
60
80
100
0-20
21-40
41-60
61+
Categorie varsta


Corespondenta
Sex
female
male
Nr persoane
0
100
200
300
400
500
Capt
Col
Don
Dr
Jonkheer
Lady
Major
Master
Miss
Mlle
Mme
Mr
Mrs
Ms
Rev
Sir
Countess


Relația dintre tarif, clasă și supraviețuire
Tarif
250
200
150
100
50
0
1
2
3
Clasă
Survived
0
1