SERBAN Alexandru Dumitru - 314CD

Cerinta 1:

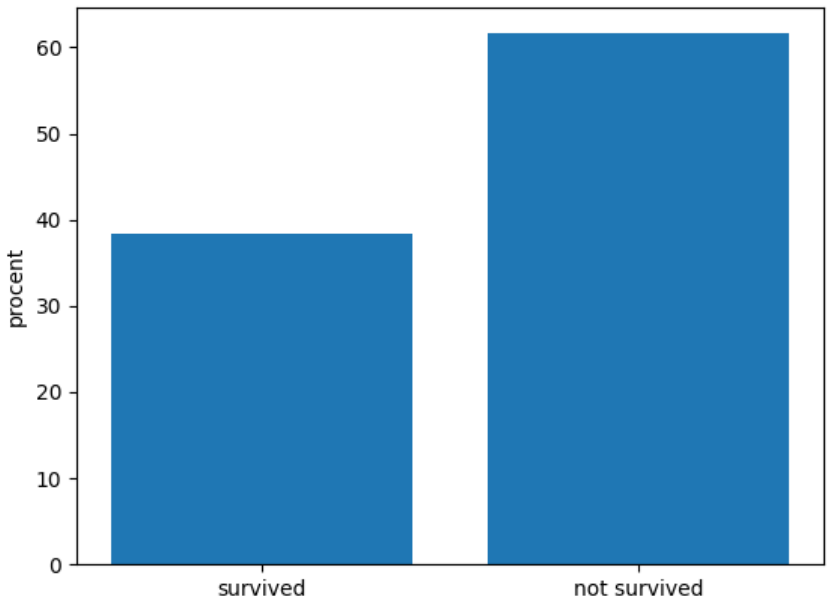
Pentru aflarea numarului de linii si de coloane am folosit df.axes[0] si df.axes[1] care imi returneaza headerele pe linii si coloane. Pentru tipurile coloanelor am folosit .dtypes. Pentru numarul de valori lipsa am folosit metoda isna care imi da valori booleene true daca valoarea este lipsa, false altfel. Insumam valorile cu metoda sum, fiind 1 fiecare valoare lipsa, ne rezulta numarul de valori lipsa. Pentru a afla daca avem linii duplicate folosim metoda duplicated care da o lista de booleene cu true daca linia este duplicata. Deci le insumam pe toate si daca rezulta o valoare mai mare ca 0 inseamna ca avem cel putin o linie duplicata.

Cerinta 2:

Pentru a determina procentele cerute numaram mereu elementele din fiecare categorie si impartim la numarul de linii, adica numarul total de inregistrari astfel :

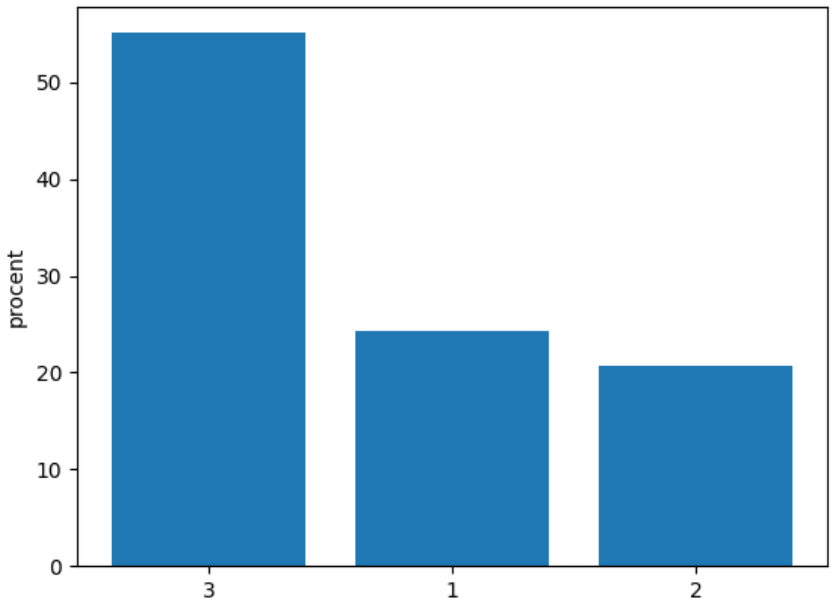
- pentru procentul de persoane care au supravietuit numaram persoanele care au supravietui si nu si aflam procentele. Cream graficul cu survived si not survived pe axa Ox si procentul de supravietuire pe axa Oy folosint plt.bar din biblioteca matplotlib.pyplot, plus alte modificari.

  Grafic :



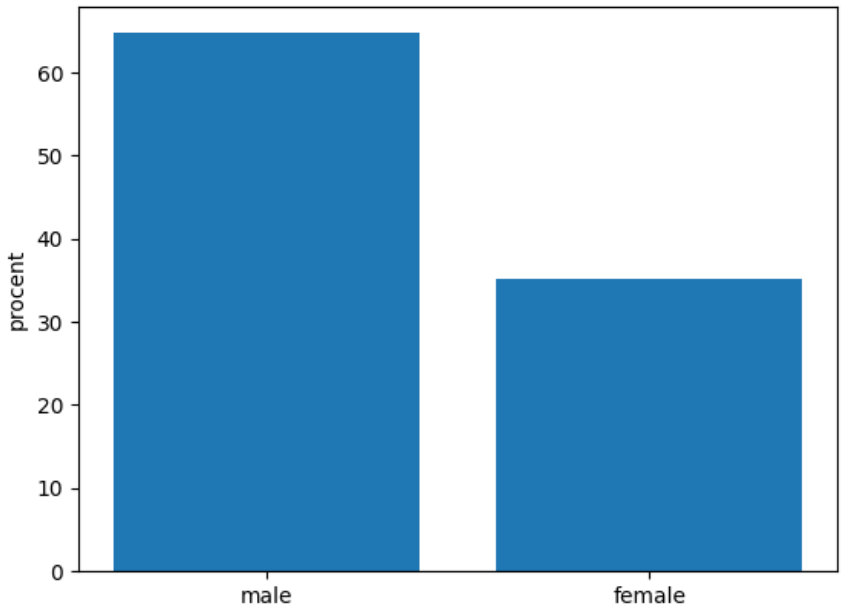
- pentru procentul din fiecare clasa am creat un dictionar ce are pe post de chei clasele si pe post de valori numarul de aparitii ale fiecarei clase. Similar cream graficul folosind pe Ox cheile si pe Oy valorile, ca procent. Transform cheile in striguri pentru a le arata separat deoarece fiind numerice daca il las default graficul nu mai descrie asa clar ca pe axa Ox este o CLASA.

Grafic:



- pentru procentul de femei, barbati analog cu procentul de supravietuitori

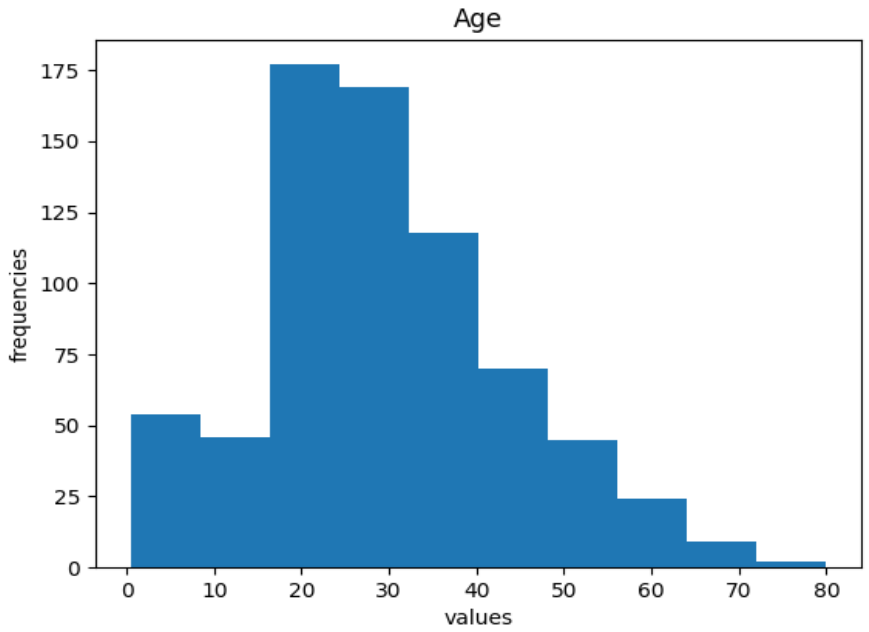
Grafic:

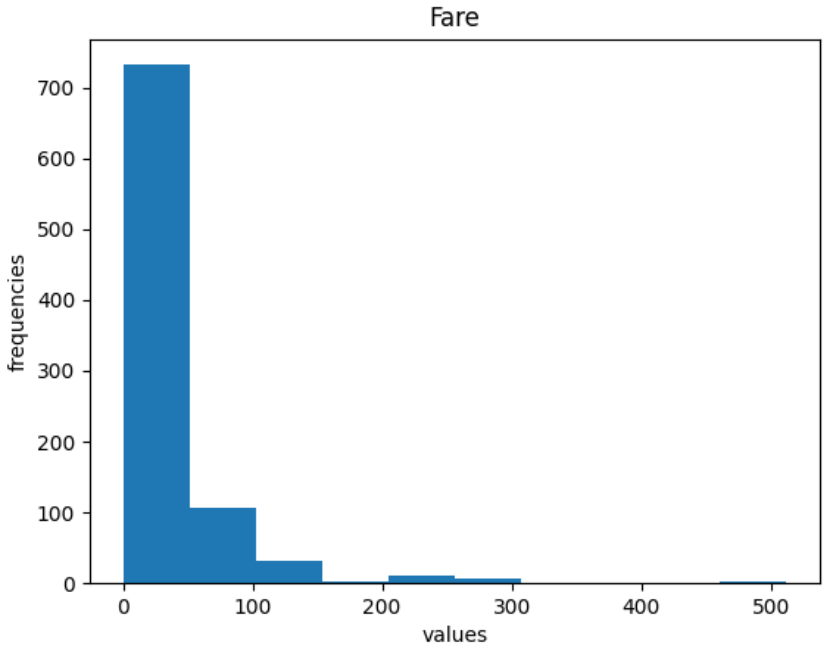


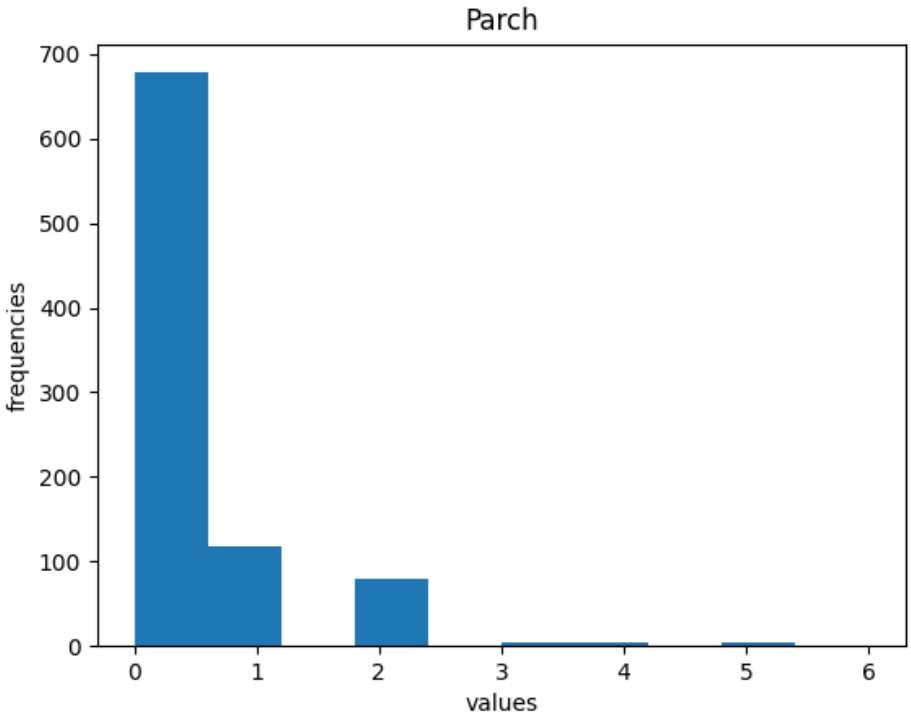
Cerinta 3:

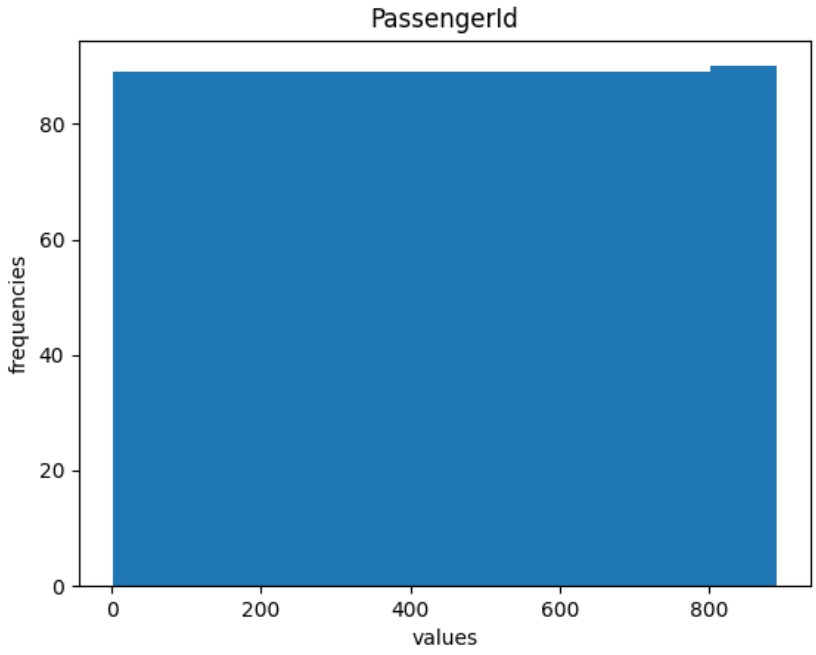
Pentru a afisa histogramele coloanelor numerice doar iteram prin coloane, le verificam tipul de date folosind dtypes si daca este numeric (int sau float). Folosim plt.hist cu coloana ca parametru care va pune pe axa Ox toate valorile diferite din coloana si pe axa Oy numarul de repetari ale fiecarei valori.

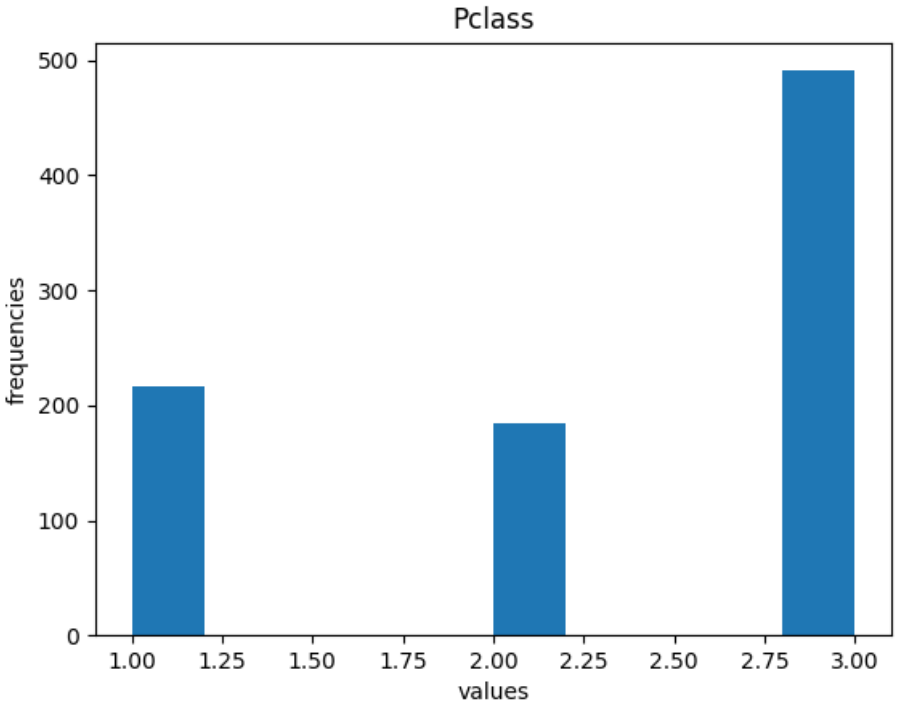
Histrogramele:

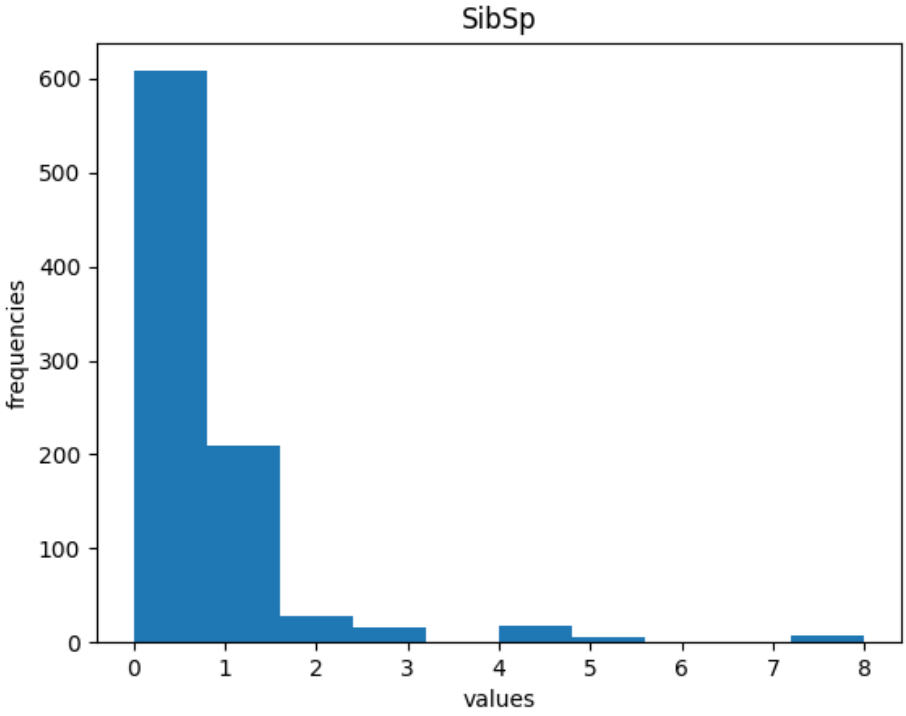


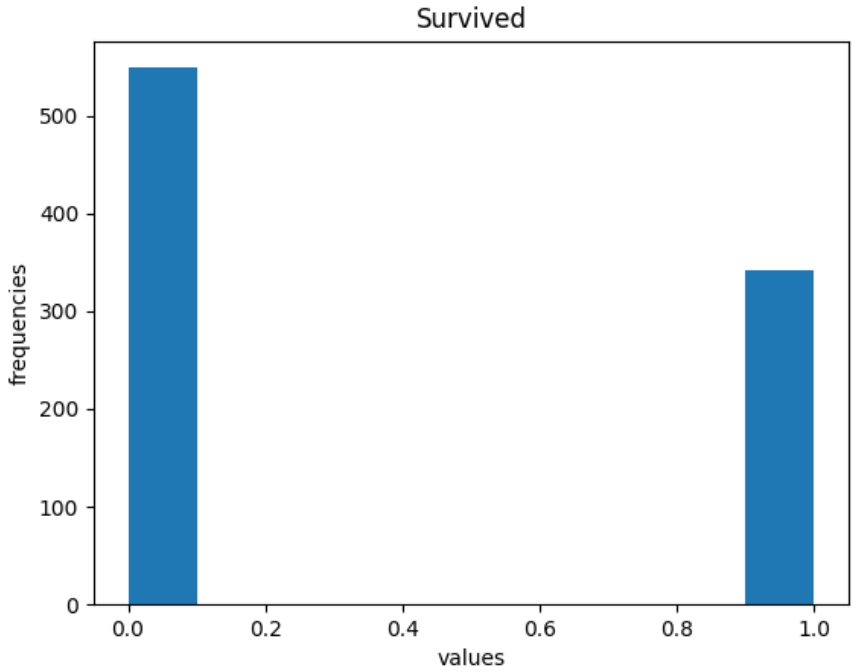



Parch
700
600
500
400
300
200
100
0
frequencies
0
1
2
3
4
5
6
values


PassengerId
80
60
40
20
0
frequencies
0
200
400
600
800
values


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


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



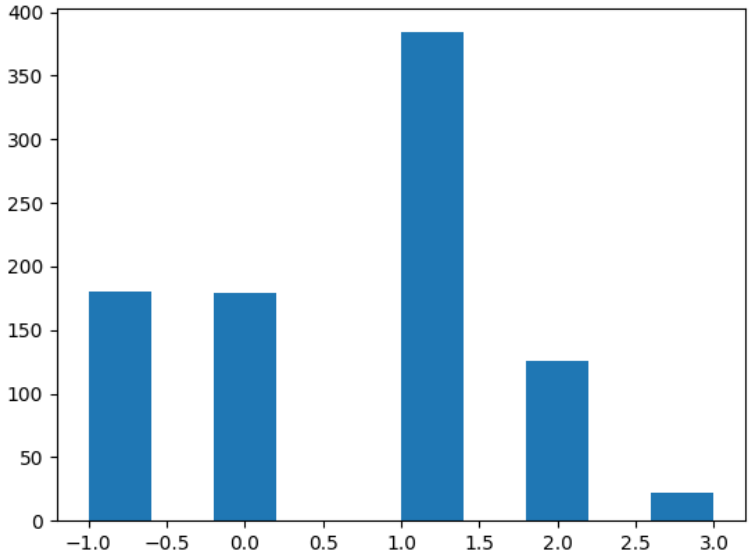
Cerinta 4:

Am iterat prin coloane si am determinat inainte daca are valori nule. Daca da calculam procentul de valori nule din toate valorile. Apoi iteram prin elementele coloanei si numaram elementele nule si nenule si care au supravietiut dintre acestea si afisam fiecare procent.

Cerinta 5:

Cream un dictionar care are pe post de chei numere de la 0 la 3 (indecsii intervalelor din cerinta) si pe post de valori numarul de pasageri cu varsta in intervalul corespunzator. Cream coloana noua si o inseram dupa coloana cu age, dupa ce ii determinam indicele, folosind metoda insert. Initializam coloana cu valori de -1, deci pe randurile in care va ramane -1 va insemna ca nu avem disponibila varsta pasagerului. Verificam fiecare varsta si punem pe linia intervalul corespunzator si incrementam valoarea in dictionar. Salvam noul dataframe. Cream apoi histograma dand ca parametru coloana de intervale care va afisa numarul de persoane in fiecare interval, pentru x-ul -1 inseamna vom avea numarul de persoane carora nu le cunoastem varsta.

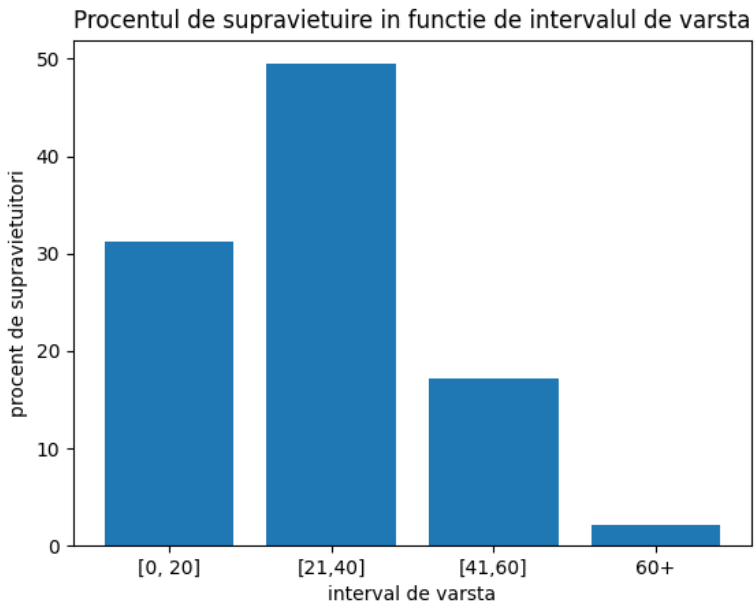
Histograma:



Cerinta 6:

Pentru a calcula procentul de supravietuire in functie de intervalele de mai devreme citim din fisierul csv creat, si din nou, cream un dictional cu cheile ce reprezinta indicele intervalelor. Iteram prin elementele coloanei de intervale si numaram fiecare interval de cate ori apare, dar si cator barbati le cunoastem varsta. Facem acest lucru, deoarece vrand sa calculam un procent, nu vom considera numarul total ca numarul total de barbati deoarece nu iei in calcul ceea ce nu cunosti intr-un studiu, deci luam ca numar total numarul total de barbati pentru care avem varsta.

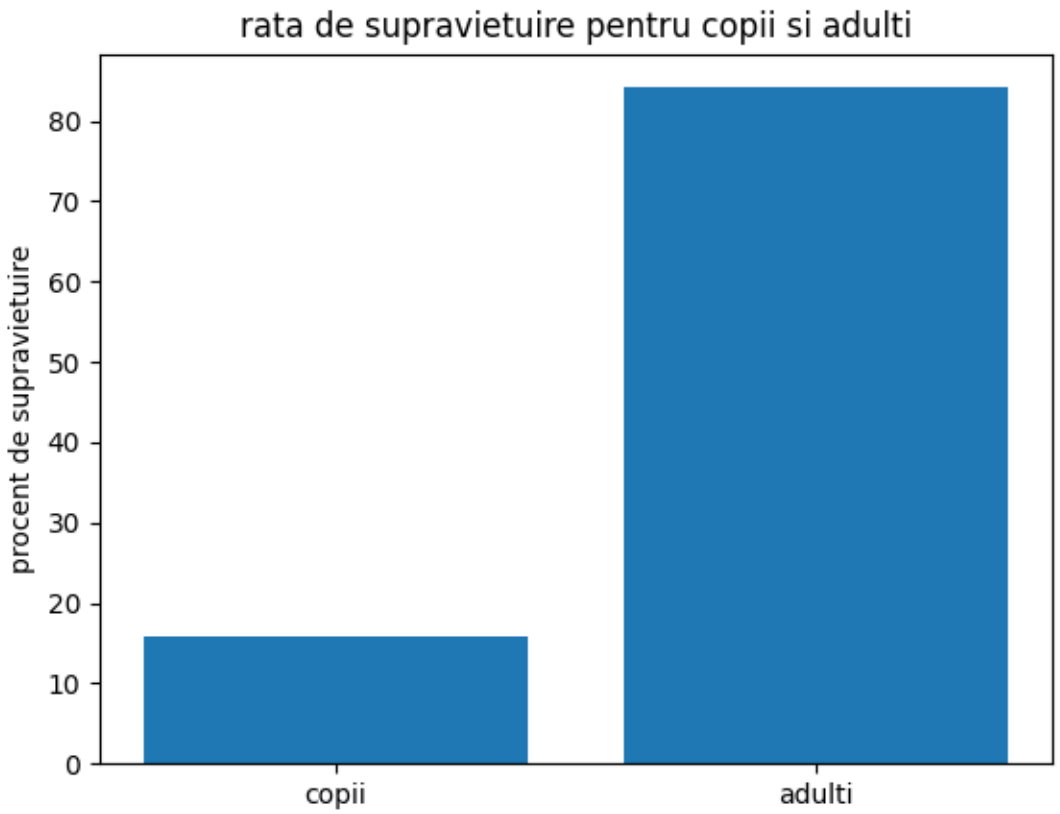
Grafic:



Cerinta 7:

Numaram toti copii, adica persoane cu varsta intre 0 si 18 ani, si toti adultii, adica celelalte persoane si, ca mai devreme, numaram si toate persoanele carora cunoastem varsta pentru a calcula un procent cu sens. Realizarea graficul este foarte similar cu cerinta 1.

Grafic:



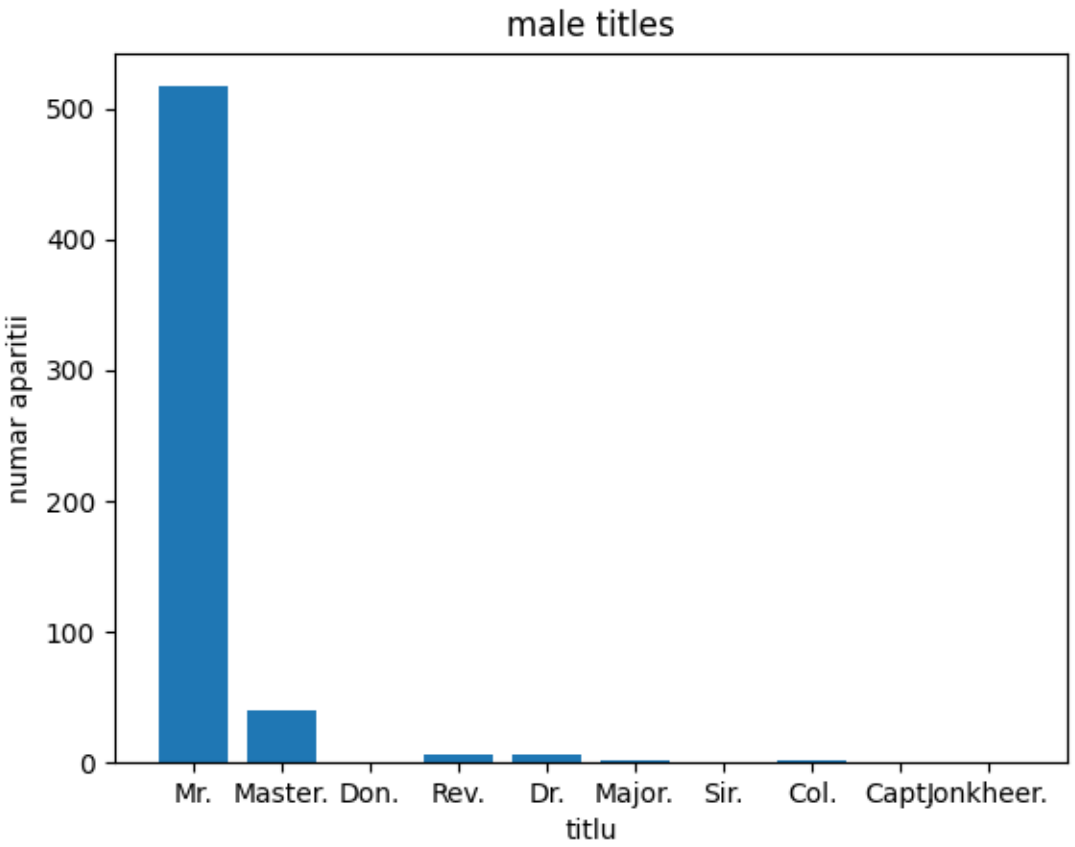
Cerinta 8:

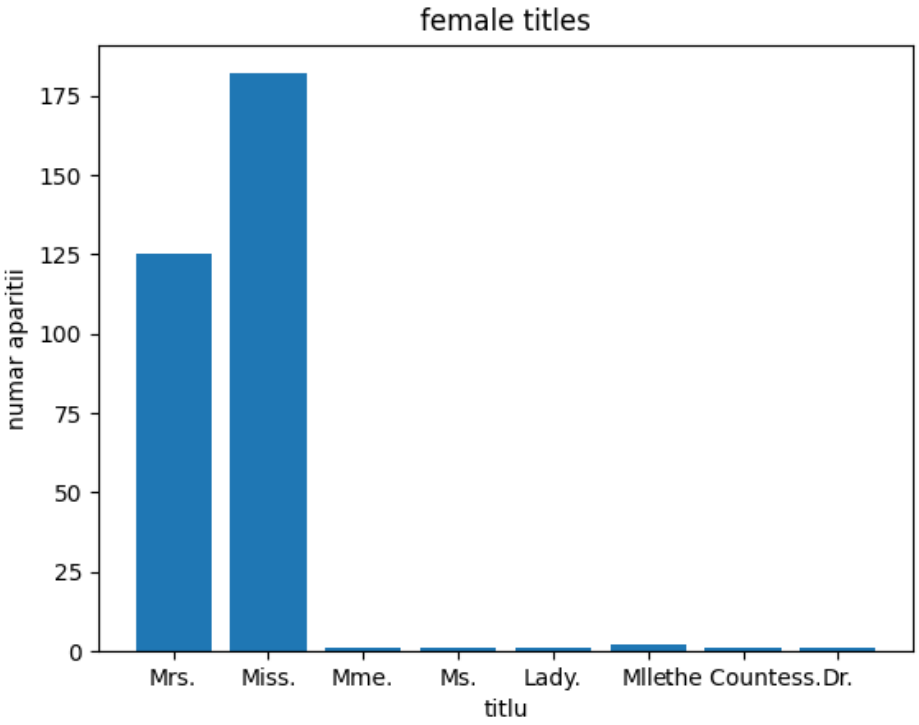
Cerinta 8 a avut un enunt putin mai neclar pentru mine, anume cum stim pentru ce coloane avem de facut media si pentru ce coloane avem de calculat frecventa. Am considerat, ruland cerintele anterioare si vazand ca singurele coloane ce au lipsuri sunt Age, Cabin, Embarked, ca daca avem tip de date int sau float vom face media, in caz contrar calculam frecventa. Acest lucru nu este corect in general, deoarece de exemplu pentru coloana Survived, chiar daca este de tip int nu ar trebui sa calculam media, deoarece o persoana nu poate supravietui 0,7.. Continuand ideea de mai sus, daca tipul coloanei este int numaram cati sunt in viata si nu, si adunam valorile pentru fiecare categorie, calculam media si apoi o punem in locurile lipsa. Pe cazul non-numeric folosim iarasi un dictionar de frecventa, aflam valoarea cu numar maxim de aparitii si o punem pe toate valorile lipsa. Salvam noul fisier csv.

Cerinta 9:

La cerinta 9 iarasi am avut neclaritati, de data aceasta mai mari. Am tratat enuntul in felul urmator: am numarat pentru fiecare gen numarul de aparitii al fiecarui titlu corespunzator genului si creat cate o histograma care sa arate frecventa numarului de aparitii. Iarasi, am folosit 2 dictionare de frecventa, unul pentru titlurile de barbati si unul pentru titlurile de femei. In extragerea titlului am folosit regex (am importat re), am observat in fisier ca pentru a gasi titlul din nume trebuie sa fie precedat de ", " si nu mai contina "." sau spatiu pana la punctul de final. Am adaugat in dictionar, ca pana acum dar am uitat sa mentionez, astfel : incrementez numarul de aparitii al cheii, iar pentru exceptia in care nu am deja in dictionar cheia adaug cheia cu valoarea 1. Creez apoi graficele pe baza perechilor cheie valoare din acest dictionar.

Grafice:

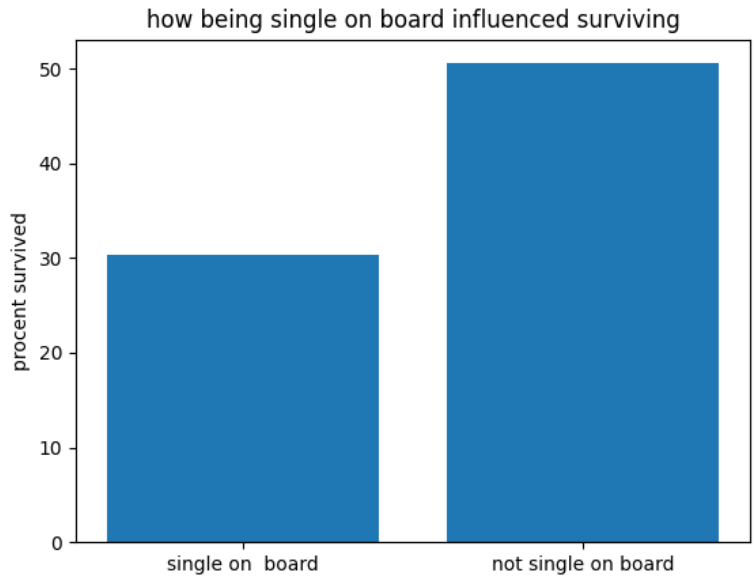




Cerinta 10:

Pentru prima parte, graficul a fost ca in cerintele de pana acum: am numarat persoanele care nu au rude la bord (adica SibSp - rude + Parch - copii == 0) si care au rude, si cei care au supravietuit dintre acestia. Am contruit graficul pe baza acestor informatii.

Grafic:



In continuare am folosit sb.catplot pentru a crea graficul cu urmatorii parametrii : data = df.head(100) pentru a lua ca baza de date primele 100 de inregistrari din dataframe ul nostru; ca axe Ox si Ox tariful si clasa, si am folosit starea de supravietuire pe post de culoare, fiind booleana avem doar 2 culori; kind = swarm pentru a nu se suprapune punctele; aspect = 3 pentru a mari imaginea ca sa aiba loc toate punctele.

Grafic: