

Nita-Pirvu George-Andrei 311CA

Partea 1

Cerinta 1

Output:

Number of columns: 12

Data types:

```
PassengerId    int64
Survived        int64
Pclass          int64
Name            object
Sex             object
Age            float64
SibSp           int64
Parch           int64
Ticket          object
Fare            float64
Cabin           object
Embarked        object
dtype: object
```

Missing values:

```
PassengerId    0
Survived        0
Pclass          0
Name            0
Sex             0
Age            177
SibSp           0
Parch           0
Ticket          0
Fare            0
Cabin           687
Embarked        2
dtype: int64
```

Number of rows: 891

Duplicates: 0

Implementare: A fost foarte usor, date fiind functiile deja existente care rezolva cerinta.

Cerinta 2:

Output:

How many survived: 38.38383838383838%

How many did not survive: 61.61616161616161%

Passengers Class:

3: 55.106621773288445%

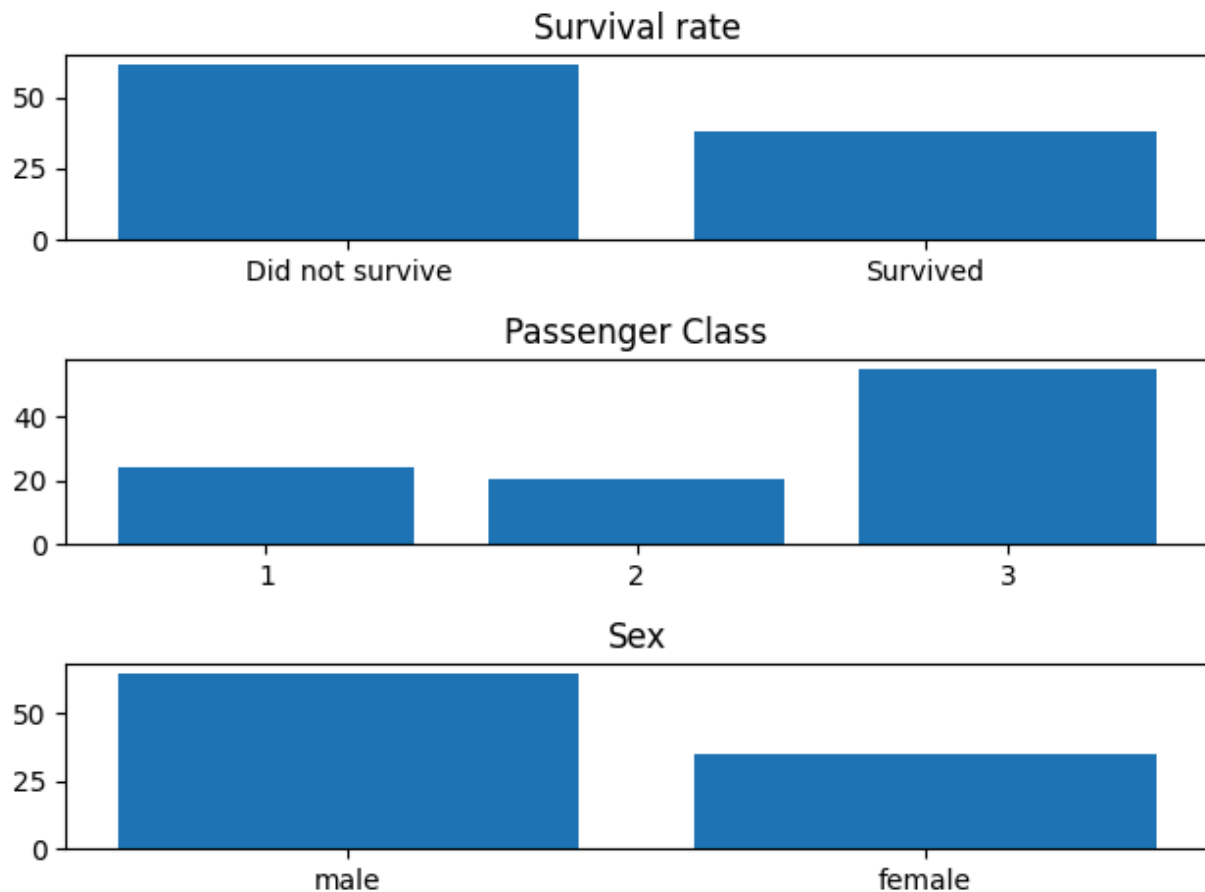
1: 24.242424242424242%

2: 20.65095398428732%

Sex:

male: 64.75869809203144%

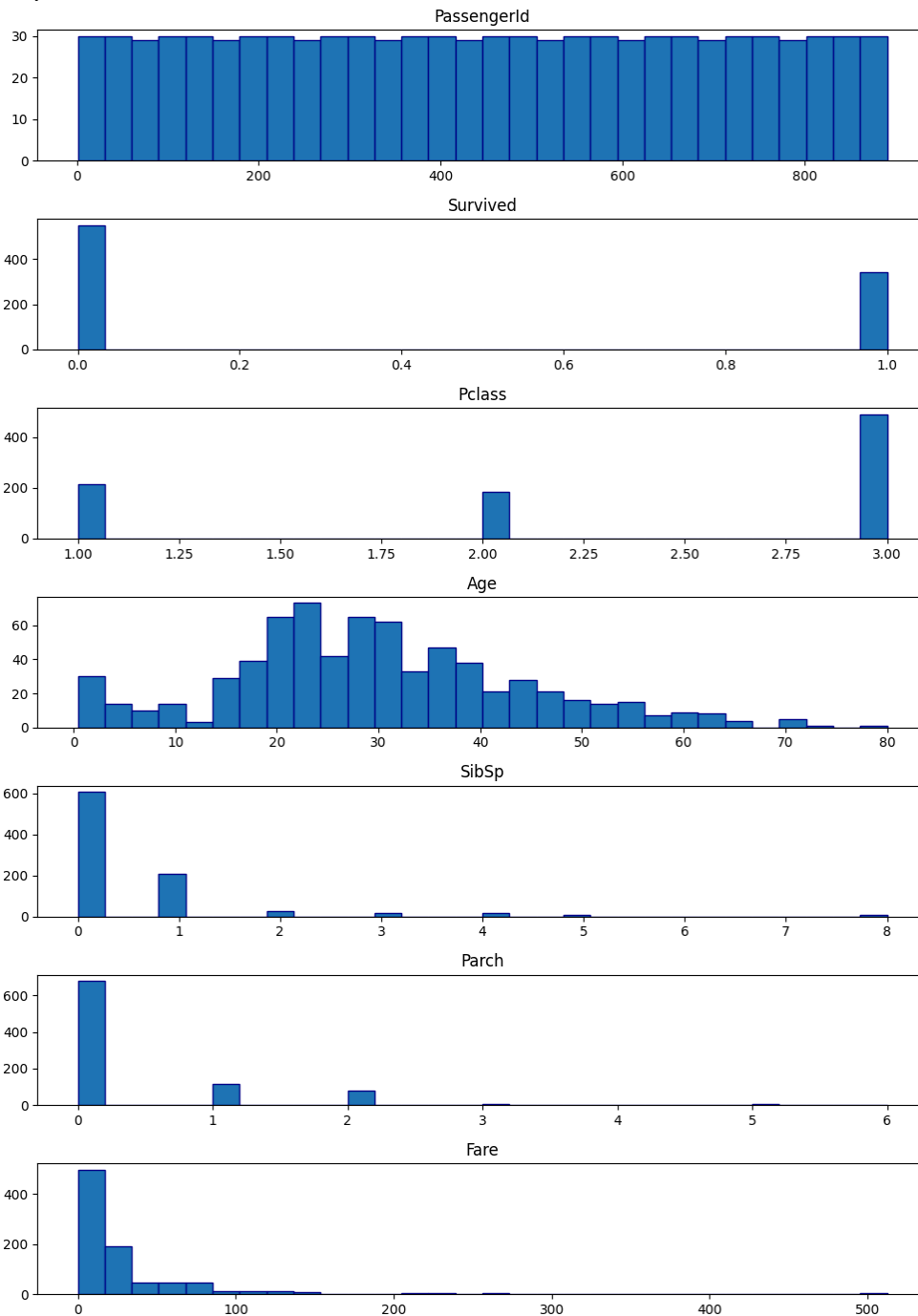
female: 35.24130190796858%



Ne folosim de `value_counts` cu parametrul `normalize = True` pentru a obtine imediat valorile de care avem nevoie, doar ca in intervalul `[0, 1]`. Mai trebuiesc inmultite cu 100 si avem procentul.

Cerinta 3

Output:



Fiind vorba doar de generarea unei histograme, singura provocare aici este dimensiunea potrivita, pe care am gasit-o prin trial & error.

Cerinta 4:

Output:

Column: Age

Missing Values: 177, %: 19.865319865319865%

Column: Cabin

Missing Values: 687, %: 77.10437710437711%

Column: Embarked

Missing Values: 2, %: 0.22446689113355783%

Survived:

Column: Age

Missing Values: 52, %: 15.204678362573098%

Column: Cabin

Missing Values: 206, %: 60.23391812865497%

Column: Embarked

Missing Values: 2, %: 0.5847953216374269%

Did not survive:

Column: Age

Missing Values: 125, %: 22.768670309653917%

Column: Cabin

Missing Values: 481, %: 87.61384335154827%

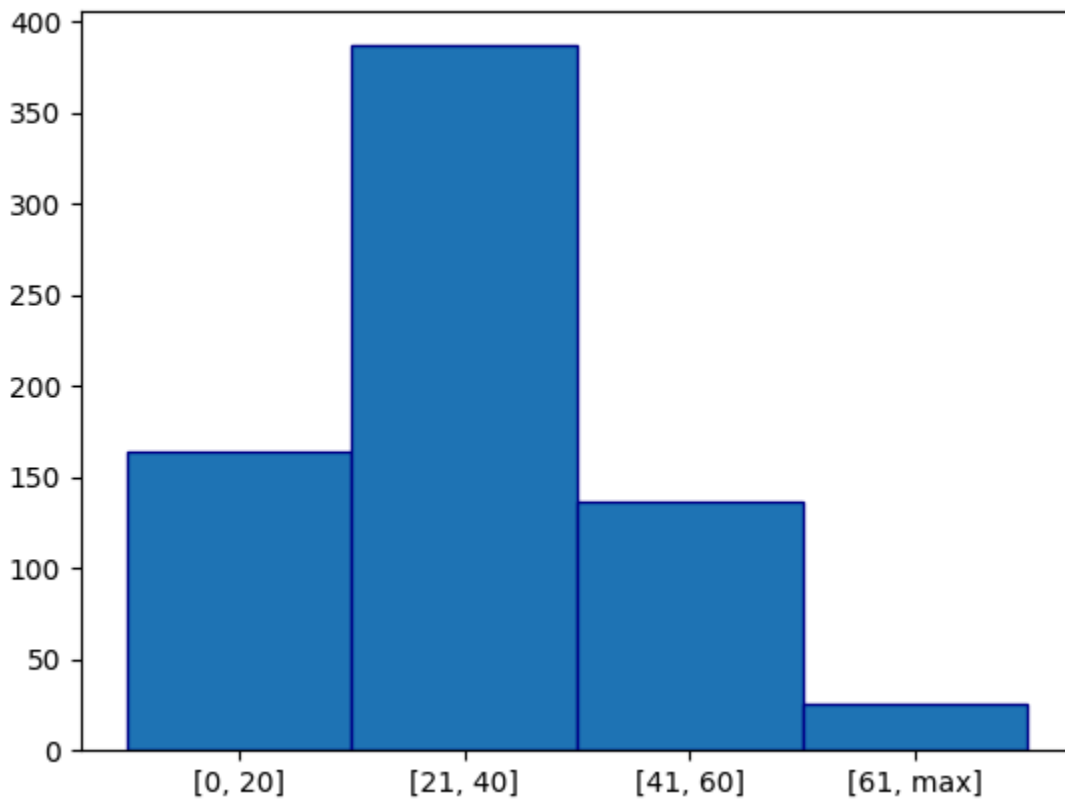
Column: Embarked

Missing Values: 0, %: 0.0%

Pentru a filtra datele aici, folosim `data['Survived'] == 1`, care ne va pastra doar intrarile care au `'Survived' = 1`.

Cerinta 5

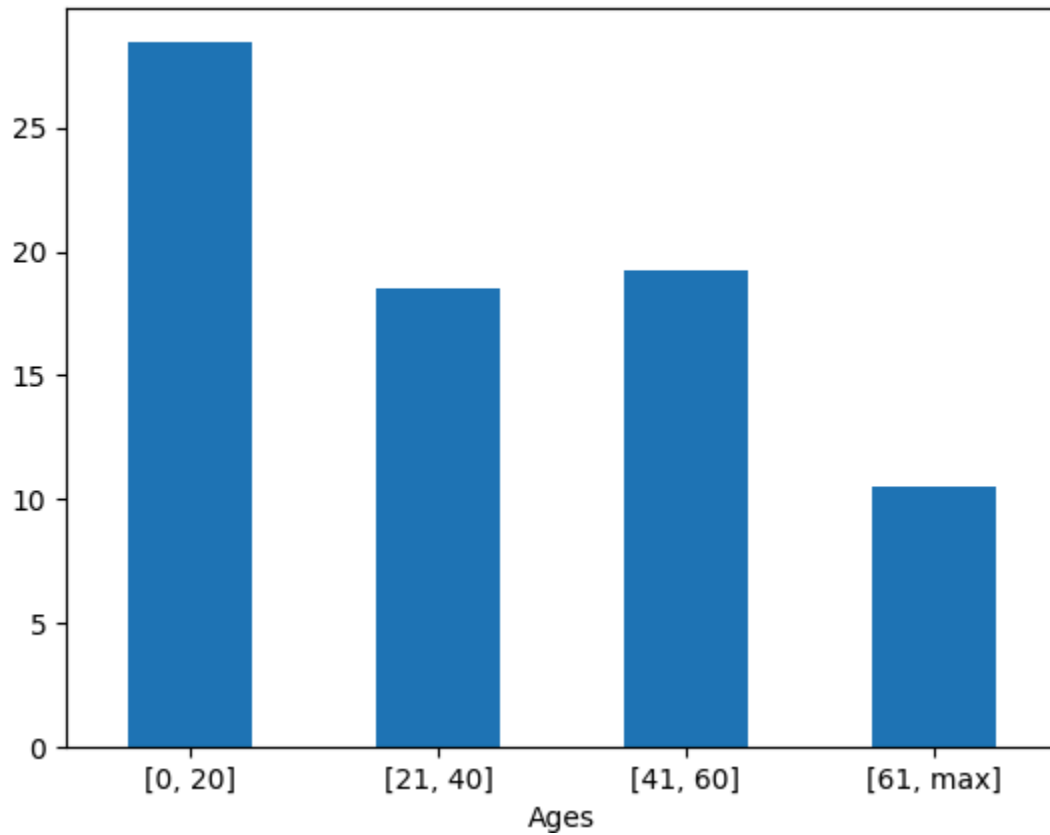
Output:



Aici, dificultatea vine din centrarea datelor pe grafic (adica, spre exemplu, bin-ul [0,20] sa fie centrat in 10, nu in 0 sau 20). Pentru asta, am definit un offset, pe care il vom adauga la inceputul fiecarui bin. In rest, implementarea nu aduce alte provocari, dat fiind faptul ca `pd.cut` ne va rezolva aproape toate cerinta.

Cerinta 6:

Output:



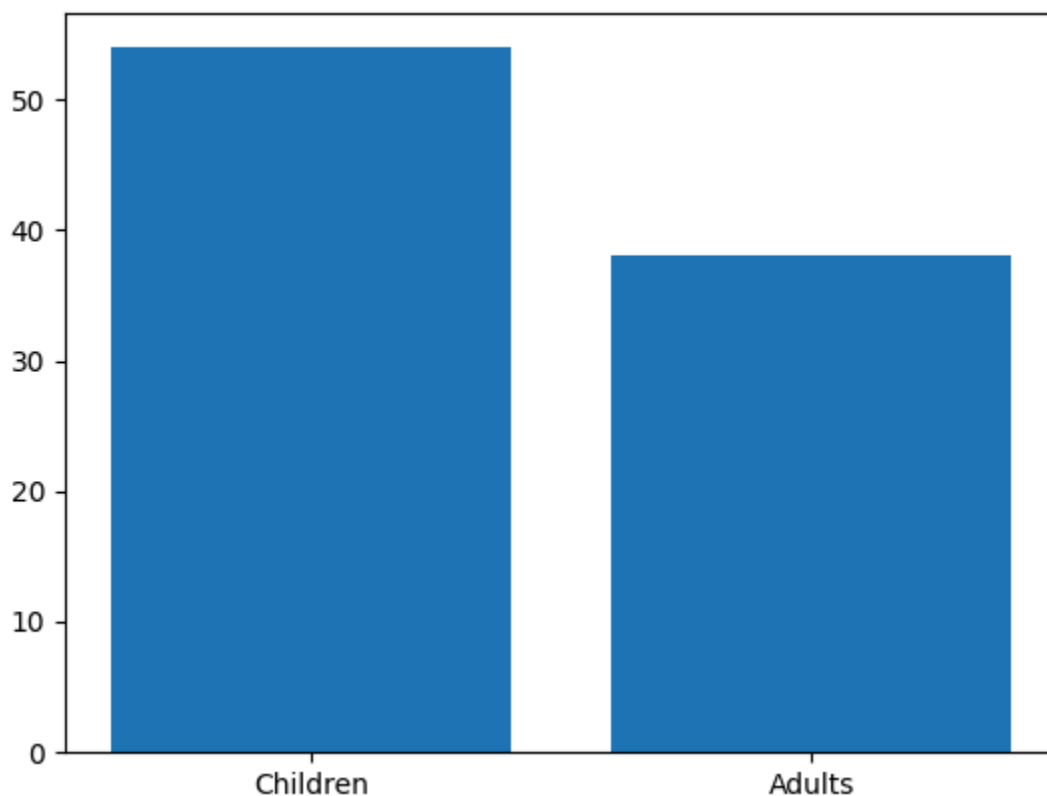
Aici, mai intai selectionam dupa sex. Salvam aceasta data, apoi in functie de Survived. Acum, le vom compara. Pentru a nu se roti label-urile, folosim `plt.xticks(rotations=0)`.

Cerinta 7:

Output:

How many children survived: 53.98230088495575%

How many adults survived: 38.10316139767055%



Aici, m-am bazat pe faptul ca Survived este 1 sau 0, practic valoarea mean reprezinta fix media supravieturii, care mai trebuia inmultita cu 100.

Cerinta 8:

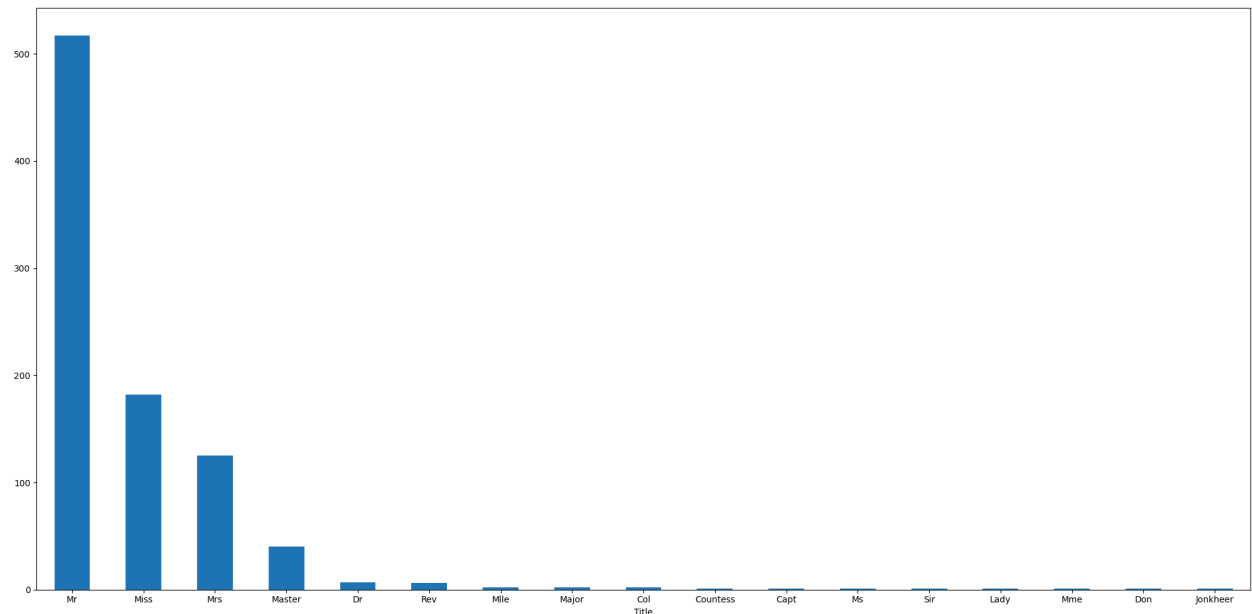
Output:

DATE_TASK8.csv

Aici se va comenta linia corespunzatoare pentru a salva datele in fisierul "DATE_TASK8.csv". Am luat fiecare field numeric si am completat cu valoarea sa medie, apoi le-am luat pe cele care sunt "obiect": nu sunt neaparat valorile numerice simple. Aici, am luat cea mai comuna inregistrare si am pus-o in locul celor absente.

Cerinta 9:

Output:



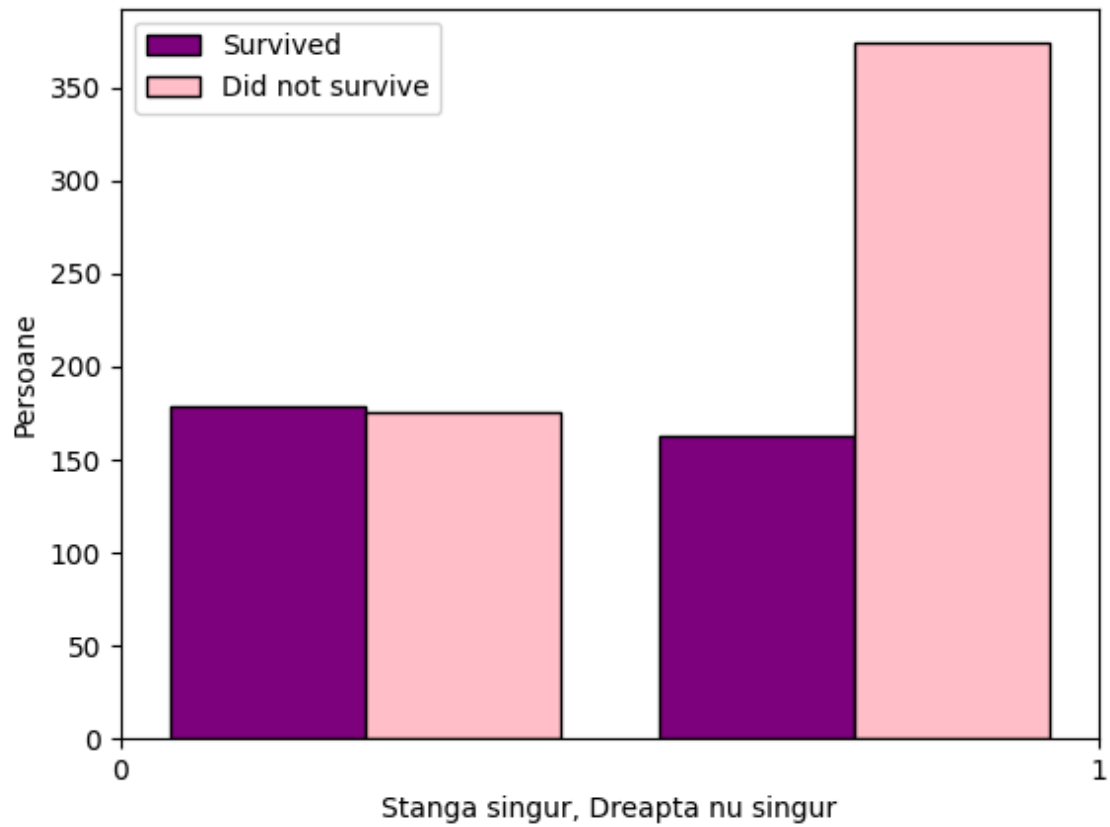
Am definit functia `check_title` care va verifica daca in numele unei persoane se afla titlurile.

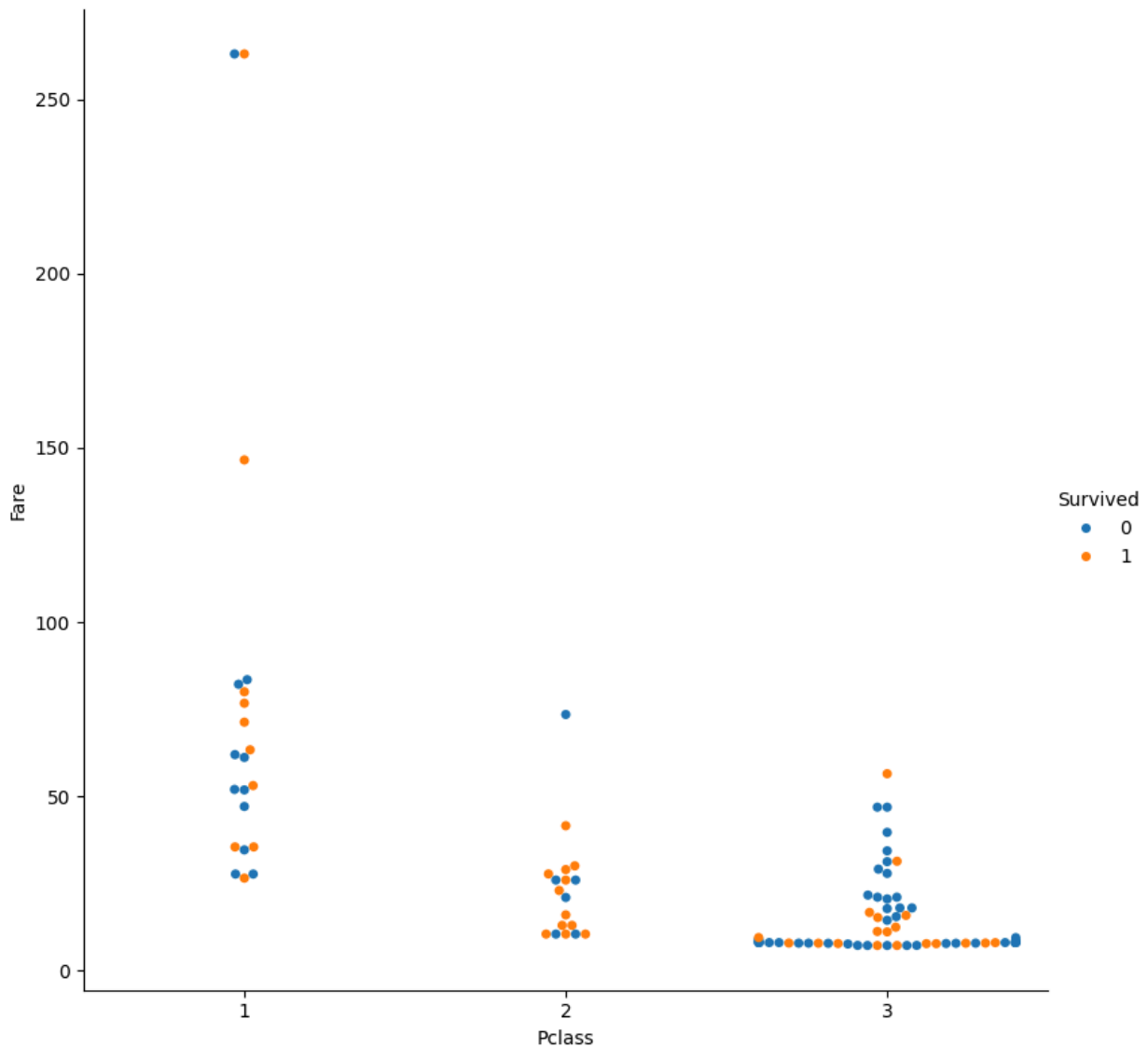
Apoi, va compara daca titlurile gasite sunt pentru genul corespunzator.

Am avut nevoie si de `apply` pentru a adauga coloane "`Check_title`", in care se justifica daca titlul este atribuit corect.

Cerinta 10:

Output:





Aici, am folosit catplot pentru a realiza ultimul grafic. Astfel, într-un grafic bidimensional, am reprezentat 3 tipuri de intrari.

Da, a fi singur a influențat rata de supraviețuire.

Majoritatea pasagerilor din clasa 3, care au plătit tarife mai mici, nu au supraviețuit.

În general, tarifele mai mari sunt asociate cu o clasă socială mai înaltă și cu șanse mai mari de supraviețuire, în special pentru clasa 1.