README

Link github

Taskul 1:

Ne folosim de libraria Pandas pentru a citii datele din fisierul de tip CSV din folderul titanic si le stocam intr-un data frame numit df.

```
df = pd.read_csv(r'titanic\train.csv')
```

Pe prima linie este afisat numele taskului urmat de numarul de coloane calculat cu ajutorl lui shape[1].

```
for i in range(0,df.shape[1]):
    print(df.columns[i],end="  ")
```

Ciclam prin fiecare coloana si afisam numele sau separate prin spatiu, iar pe urmatoarea line se afiseaza in aceeasi ordine tipul de date (df.dtypes.iloc[i]). Rezultatul avand formatul

Nume coloana

Tip coloana

Urmeaza sa ne folosim de isna() (is not a) pentru a determina elementele cu valori lipsa .

Numarul de linii duplicate se calculeaza prin diferenta dintre numarul total de linii si nr de linii dupa eliminarea celor duplicate cu ajutorul lui drop_duplicates

```
print("Numarul de linii dublicate este " + str(len(df.to_string()) -
len(df.drop_duplicates().to_string())))
```

Taskul 2:

Procentul supravietuitorilor se calculeaza prin selectarea coloanei Survived si folosindu ne de mean() calculam media iar inmultita cu 100 ne da procentul

```
survivors = int(df["Survived"].mean()*100)
```

Cel al mortilor se scade din 100% pe cei vii

```
deads = 100 - survivors
```

Asemanator se calculeaza si femeile cu barbatii

Pentru charturi se foloseste pie

```
graph, pies = plt.subplots(1, 3)

pies[0].pie(data[0:2], labels=data_names[0:2], autopct='%1.1f%%')
pies[1].pie(data[2:4], labels=data_names[2:4], autopct='%1.1f%%')
pies[2].pie(data[4:], labels=data_names[4:], autopct='%1.1f%%')
plt.show()
```

luand datele dfin vectorul de data_names ca labeluri si datele din data.

Taskul 3:

Se concateneaza toate datele din datafileurile din Titanic intr-un file df complet dupa care se aplica operatii pe acesta.

Pentru fiecare coloana care nu e passanger id si care are valoare numerica se construieste o histograma astfel:

Pe axa orizontală sunt incluse intervalele de valori ale variabilei, iar pe axa verticală se reprezintă numărul de exemple din setul de date care sunt incluse în fiecare interval.

Taskul 4:

Se identifica coloanele cu valori lipsa prin

```
isna()
```

filtram coloanele din df cu valori lipsă, rezultând un dataframe df_col_lipsa care conține aceste coloane.

```
df_col_lipsa = df.loc[:, col_lipsa]
```

procentele de supraviețuitori și non-supraviețuitori care au date incomplete

```
surv = surv  + (df_col_lipsa[col].isna() & df['Survived']).mean() #facem & inte valorile nan care sunt true si coloana de survived 1 deci ne dau pers supravietuitoare cu incomplete
    dead = dead  + (df_col_lipsa[col].isna() & (df['Survived']==0)).mean()
```

Taskul 5:

Presupune categorizarea vârstei pasagerilor, crearea unei noi coloane pe baza acestei categorizări, și vizualizarea datelor printr-un grafic.

```
num_cat_varsta = [(df["Age"] <= 20).sum(),(((df["Age"] > 20)& (df["Age"]<=40)).sum()),((df["Age"] > 40)& (df["Age"]<=60)).sum(),(df["Age"] > 60).sum()]
```

Taskul 6:

Facem o coloana doar pentru barbati dupa iar o filtram pt cei care au supravietuit si o grupam dupa categoria de varsta, urmand a face un graphic pentru ea

```
df_male = df[df['Sex'] == 'male']
df_male_survived = df_male[df_male["Survived"] == 1]
```