Лабораторна робота №4.
Візуальний аналіз даних про пасажирів Титаніку

Заповніть код в клітинках замість "Ваш код тут"

In [8]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

Зчитуємо навчальну вибірку.

Доступ до даних на google drive, якщо ви відкриваєте блокнот в google colab, а не на PC, можна отримати шляхом монтування google drive

In [2]:
```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly&response_type=code (https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly&response_type=code)

Enter your authorization code:
4/4AFiQopV2FYEYPb9LV3UhLMpI4t90hb26RxHSU6CshxfBZFtb4HldL0
Mounted at /content/gdrive
```

In [3]:
```
!ls gdrive/'My Drive'/TEACHING/IntroDataScience/intro_to_data_science/Lab_3_4/da
```

```
howpop_train.csv  titanic_train.csv
```

In [4]:
```
# шлях до папки з даними на моєму google drive, відредагуйте згідно вашого випад
data_folder = "gdrive/My Drive/TEACHING/IntroDataScience/intro_to_data_science/L
```

In [13]:
```
#train_df = pd.read_csv("data/titanic_train.csv", index_col='PassengerId')

train_df = pd.read_csv( data_folder+'/titanic_train.csv', index_col='PassengerId
```

In [14]:
```
train_df.head(2)
```

Out[14]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emb
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	

In [15]:
```
train_df.describe(include='all')
```

Out[15]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
count	891.000000	891.000000	891	891	714.000000	891.000000	891.000000	891	89
unique	NaN	NaN	891	2	NaN	NaN	NaN	681	
top	NaN	NaN	Masselmani, Mrs. Fatima	male	NaN	NaN	NaN	1601	
freq	NaN	NaN	1	577	NaN	NaN	NaN	7	
mean	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381594	NaN	3
std	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806057	NaN	4
min	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000000	NaN	
25%	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000000	NaN	
50%	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000000	NaN	1
75%	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000000	NaN	3
max	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.000000	NaN	51

In [16]:
```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 891 entries, 1 to 891
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Survived  891 non-null    int64  
 1   Pclass    891 non-null    int64  
 2   Name      891 non-null    object 
 3   Sex       891 non-null    object 
 4   Age       714 non-null    float64
 5   SibSp     891 non-null    int64  
 6   Parch     891 non-null    int64  
 7   Ticket    891 non-null    object 
 8   Fare      891 non-null    float64
 9   Cabin     204 non-null    object 
 10  Embarked  889 non-null    object 
dtypes: float64(2), int64(4), object(5)
memory usage: 83.5+ KB
```

Відкинемо ознаку `Cabin`, а потім – всі рядки, де є пропуски.

In [17]:
```
train_df = train_df.drop('Cabin', axis=1).dropna()
```

Побудуйте попарні залежності ознак `Age`, `Fare`, `Pclass`, `Sex`, `SibSp`, `Parch`, `Embarked` и `Survived`. (метод `scatter_matrix` Pandas чи `pairplot` Seaborn).

In [ ]:
```
# Ваш код тут
```

Як плата за білет (`Fare`) залежить від класу каюти (`Pclass`)? Побудуйте boxplot.

In [ ]:
```
# Ваш код тут
```

Такий boxplot виходить не дуже красивим із-за викидів.

Опціонально: створіть ознаку `Fare_no_out` (вартість без викидів), в якій виключається вартість, що відрізняється від середнього по класу більш ніж на 2 стандартних відхилення. Важливо: потрібно виключити викиди саме в залежності від класу каюти. Інакше виключатися будуть тільки найбільші (1 клас) і малі (3 клас) вартості.

In [ ]:
```
train_df['Fare_no_out'] = train_df['Fare']
fare_pclass1 = train_df[train_df['Pclass'] == 1]['Fare']
fare_pclass2 = train_df[train_df['Pclass'] == 2]['Fare']
fare_pclass3 = train_df[train_df['Pclass'] == 3]['Fare']
fare_pclass1_no_out = # Ваш код тут
fare_pclass2_no_out = # Ваш код тут
fare_pclass3_no_out = # Ваш код тут
train_df['Fare_no_out'] = fare_pclass1_no_out.append(fare_pclass2_no_out)\
                                             .append(fare_pclass3_no_out)
```

Яке відношення загиблих і вижившх в залежності від статі? Відобразіть з допомогою Seaborn.countplot з аргументом hue.

In [ ]:
```
# Ваш код тут
```

Яке співідношення загиблих і вижившх в залежності від класу каюти? Відобразівть за допомогою Seaborn.countplot з аргументом hue.

In [ ]:
```
# Ваш код тут
```

Як факт виживання залежить від віку пасажира? Перевірте (графічно) припущення, що молоді частіше виживали. Нехай, умовно, молоді - молодші 30 років, похилого віку – старші 60 років.

In [ ]:
```
# Ваш код тут
```