Тема 2. Визуальный анализ данных

Практическое задание. Визуальный анализ данных по пассажирам "Титаника".

Соревнование Kaggle "Titanic: Machine Learning from Disaster".

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

Считываем обучающую выборку.

```
train_df = pd.read_csv(r"C:\Users\Елена\Downloads\data\
titanic_train.csv",
                       index_col='PassengerId')

train_df.head(2)
```

```
             Survived  Pclass  \
PassengerId
1                   0       3
2                   1       1

                                                      Name     Sex
Age  \
PassengerId

1                                   Braund, Mr. Owen Harris    male
22.0
2            Cumings, Mrs. John Bradley (Florence Briggs Th...  female
38.0

             SibSp  Parch     Ticket     Fare Cabin Embarked
PassengerId
1                1      0  A/5 21171   7.2500   NaN        S
2                1      0   PC 17599  71.2833   C85        C
```

```
train_df.describe(include='all')
```

```
          Survived      Pclass                      Name   Sex
Age  \
count   891.000000  891.000000                       891   891
714.000000
unique         NaN         NaN                       891     2
```

```
NaN
top            NaN         NaN  Braund, Mr. Owen Harris  male
NaN
freq           NaN         NaN                        1   577
NaN
mean      0.383838    2.308642                      NaN   NaN
29.699118
std       0.486592    0.836071                      NaN   NaN
14.526497
min       0.000000    1.000000                      NaN   NaN
0.420000
25%       0.000000    2.000000                      NaN   NaN
20.125000
50%       0.000000    3.000000                      NaN   NaN
28.000000
75%       1.000000    3.000000                      NaN   NaN
38.000000
max       1.000000    3.000000                      NaN   NaN
80.000000

              SibSp        Parch  Ticket        Fare    Cabin Embarked
count   891.000000  891.000000     891  891.000000      204      889
unique         NaN         NaN     681         NaN      147        3
top            NaN         NaN  347082         NaN  B96 B98        S
freq           NaN         NaN       7         NaN        4      644
mean      0.523008    0.381594     NaN   32.204208      NaN      NaN
std       1.102743    0.806057     NaN   49.693429      NaN      NaN
min       0.000000    0.000000     NaN    0.000000      NaN      NaN
25%       0.000000    0.000000     NaN    7.910400      NaN      NaN
50%       0.000000    0.000000     NaN   14.454200      NaN      NaN
75%       1.000000    0.000000     NaN   31.000000      NaN      NaN
max       8.000000    6.000000     NaN  512.329200      NaN      NaN

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
```

Выкинем признак `Cabin`, а потом – все строки, где есть пропуски.

```
train_df = train_df.drop('Cabin', axis=1).dropna()
```

Постройте попарные зависимости признаков `Age`, `Fare`, `Pclass`, `Sex`, `SibSp`, `Parch`, `Embarked` и `Survived`. (метод `scatter_matrix` Pandas или `pairplot` Seaborn).

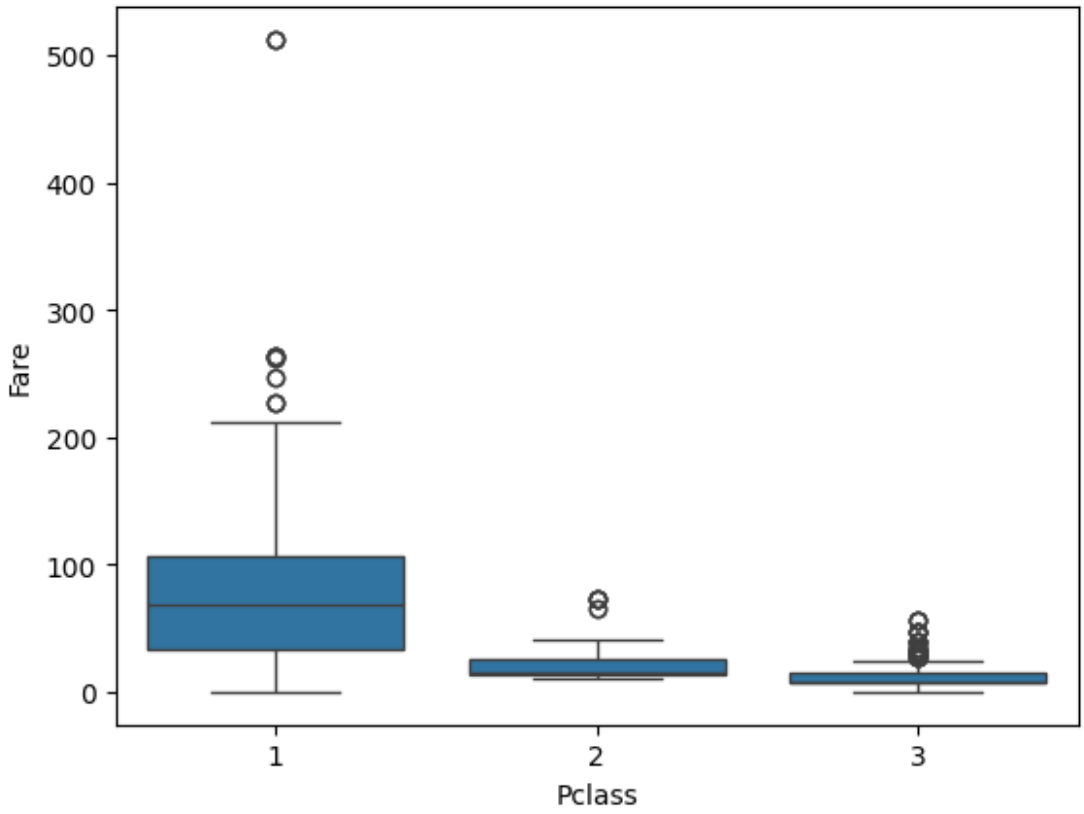
```
sns.pairplot(train_df[['Age', 'Fare', 'Pclass', 'Sex', 'SibSp',
'Parch', 'Embarked', 'Survived']],
 hue='Survived')
```

```
<seaborn.axisgrid.PairGrid at 0x21e80511b50>
```

Как плата за билет (`Fare`) зависит от класса каюты (`Pclass`)? Постройте boxplot.

```
sns.boxplot(x='Pclass', y='Fare', data=train_df)
```

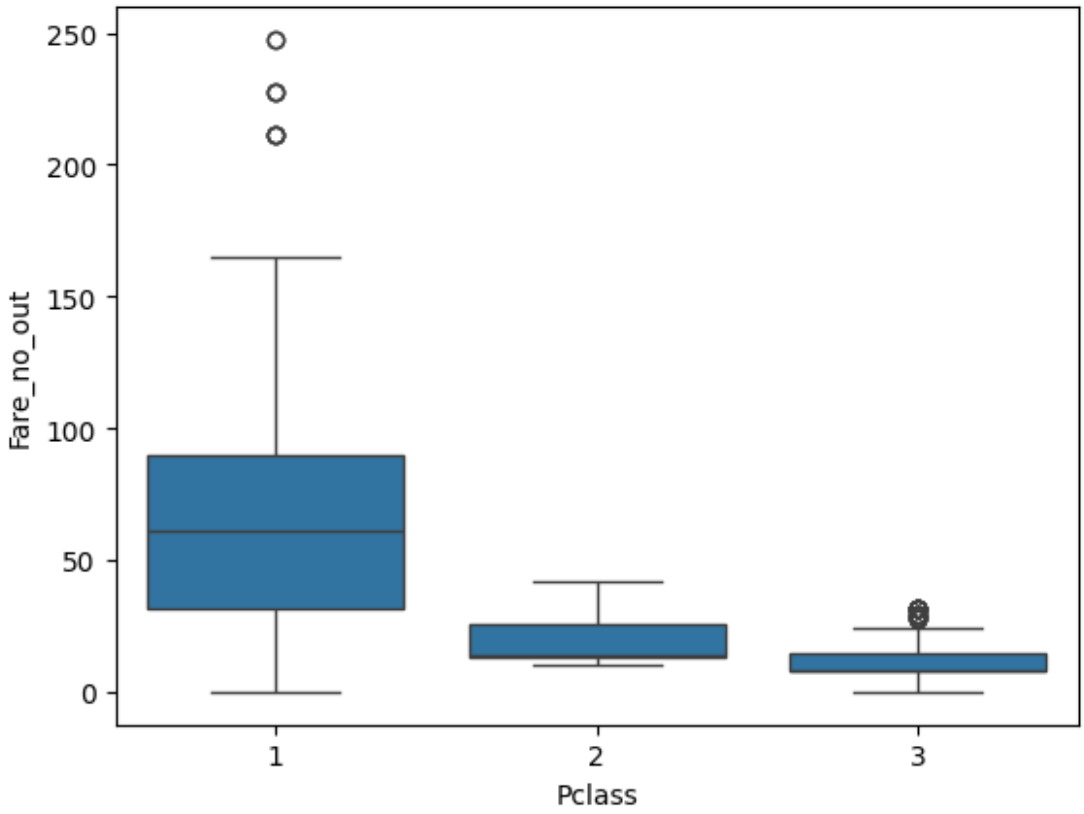
```
<Axes: xlabel='Pclass', ylabel='Fare'>
```



Такой boxplot получается не очень красивым из-за выбросов.

Опционально: создайте признак `Fare_no_out` (стоимости без выбросов), в котором исключаются стоимости, отличающиеся от средней по классу более чем на 2 стандартных отклонения. Важно: надо исключать выбросы именно в зависимости от класса каюты. Иначе исключаться будут только самые большие (1 класс) и малые (3 класс) стоимости.

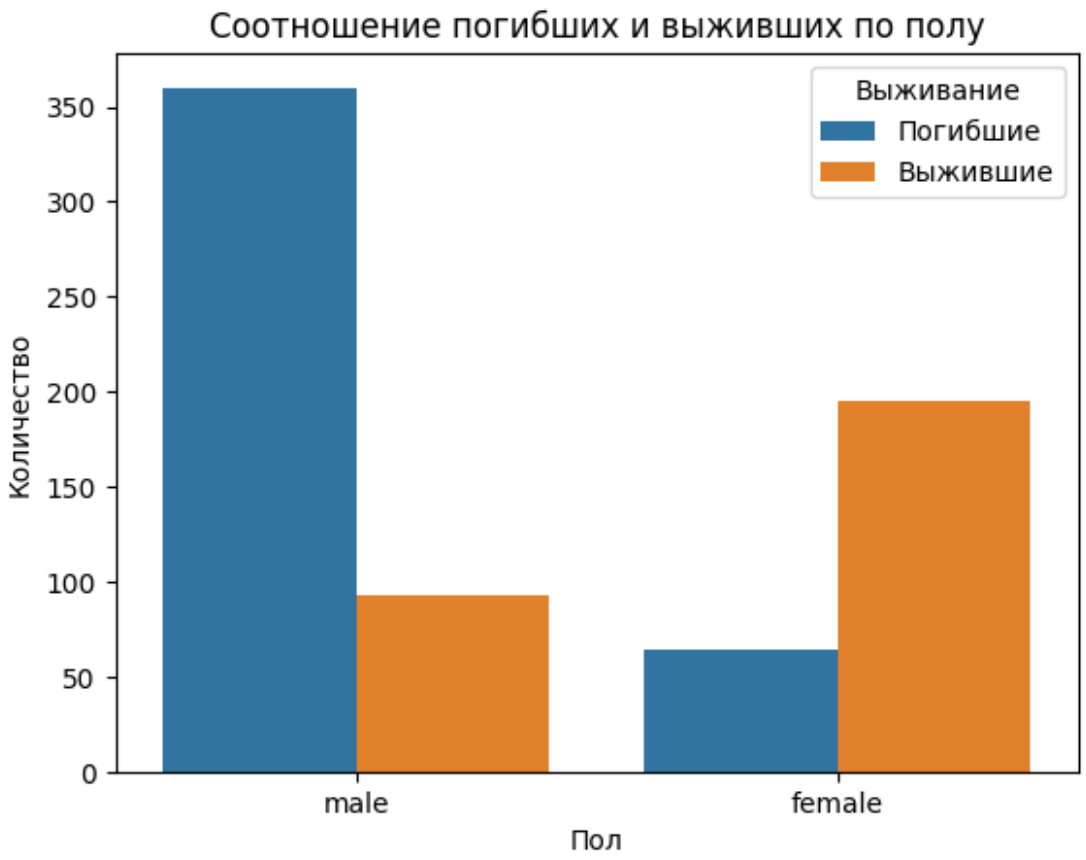
```
train_df['Fare_no_out'] = train_df['Fare']
fare_pclass1 = train_df[train_df['Pclass'] == 1]['Fare']
fare_pclass2 = train_df[train_df['Pclass'] == 2]['Fare']
fare_pclass3 = train_df[train_df['Pclass'] == 3]['Fare']
fare_pclass1_no_out = fare_pclass1[(fare_pclass1 -
fare_pclass1.mean()).abs() < 2 * fare_pclass1.std()]
fare_pclass2_no_out = fare_pclass2[(fare_pclass2 -
fare_pclass2.mean()).abs() < 2 * fare_pclass2.std()]
fare_pclass3_no_out = fare_pclass3[(fare_pclass3 -
fare_pclass3.mean()).abs() < 2 * fare_pclass3.std()]
train_df['Fare_no_out'] = pd.concat([fare_pclass1_no_out,
fare_pclass2_no_out, fare_pclass3_no_out])
sns.boxplot(x='Pclass', y='Fare_no_out', data=train_df)
plt.show()
```



Каково соотношение погибших и выживших в зависимости от пола? Отобразите с помощью Seaborn.countplot с аргументом hue.

```
sns.countplot(x='Sex', hue='Survived', data=train_df)
plt.title('Соотношение погибших и выживших по полу')
plt.xlabel('Пол')
plt.ylabel('Количество')
plt.legend(title='Выживание', loc='upper right', labels=['Погибшие',
'Выжившие'])
```

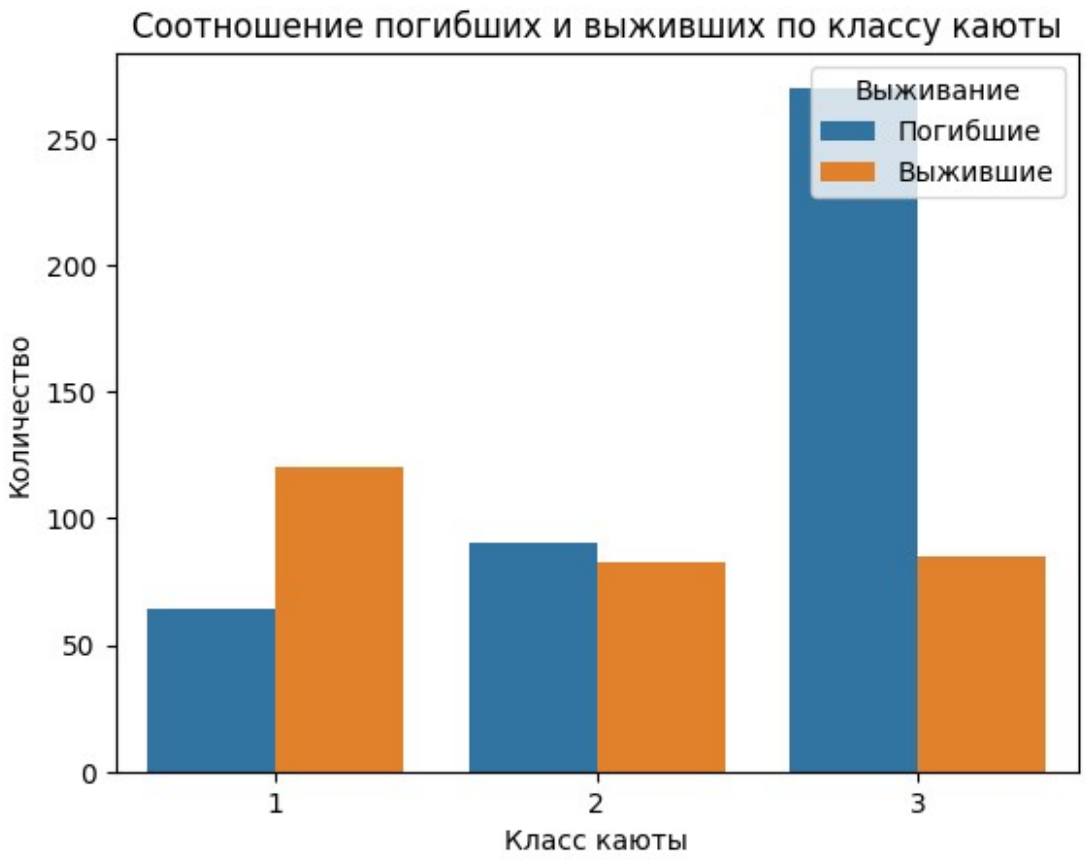
```
<matplotlib.legend.Legend at 0x21eaa86f640>
```



Каково соотношение погибших и выживших в зависимости от класса каюты? Отобразите с помощью Seaborn.countplot с аргументом hue.

```
sns.countplot(x='Pclass', hue='Survived', data=train_df)
plt.title('Соотношение погибших и выживших по классу каюты')
plt.xlabel('Класс каюты')
plt.ylabel('Количество')
plt.legend(title='Выживание', loc='upper right', labels=['Погибшие',
'Выжившие'])
```

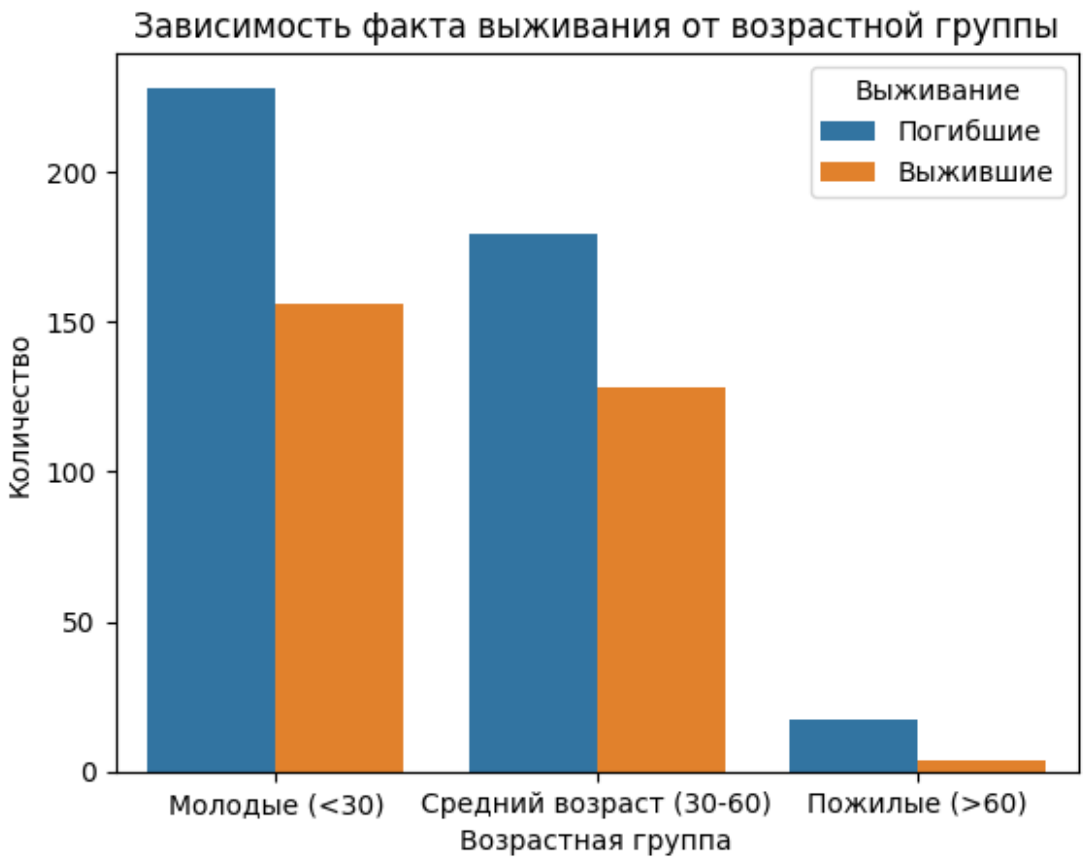
```
<matplotlib.legend.Legend at 0x21eaa9d7ee0>
```



Как факт выживания зависит от возраста пассажира? Проверьте (графически) предположение, что молодые чаще выживали. Пусть, условно, молодые - младше 30 лет, пожилые – старше 60 лет.

```
def categorize_age(age):
 if age < 30:
   return 'Молодые (<30)'
 elif age > 60:
    return 'Пожилые (>60)'
 else:
    return 'Средний возраст (30-60)'
train_df['AgeGroup'] = train_df['Age'].apply(categorize_age)
sns.countplot(x='AgeGroup', hue='Survived', data=train_df)
plt.title('Зависимость факта выживания от возрастной группы')
plt.xlabel('Возрастная группа')
plt.ylabel('Количество')
plt.legend(title='Выживание', loc='upper right', labels=['Погибшие',
'Выжившие'])

<matplotlib.legend.Legend at 0x21eaa9c3e50>
```


Зависимость факта выживания от возрастной группы
Выживание
Погибшие
Выжившие
Количество
0
50
100
150
200
Молодые (<30)
Средний возраст (30-60)
Пожилые (>60)
Возрастная группа