Разведочный анализ данных. Исследование и визуализация данных.

Текстовое описание

Я использовал дата сет Титаника, в котором описываются данные о пасажирах титаника.
Ссылка на датасет: https://www.kaggle.com/datasets/brendan45774/test-file?resource=download

Датасет состоит из одного csv-файла, "tested.csv".

Описание колонок в датасете:

- PassengerId: Уникальный идентификатор пассажира.
- Survived: Выживание (0 = не выжил, 1 = выжил).
- Pclass: Класс билета (1 = первый класс, 2 = второй класс, 3 = третий класс).
- Name: Имя пассажира.
- Sex: Пол пассажира (male = мужчина, female = женщина).
- Age: Возраст пассажира.
- SibSp: Количество братьев, сестер или супругов на борту.
- Parch: Количество родителей или детей на борту.
- Ticket: Номер билета.
- Fare: Плата за проезд.
- Cabin: Номер каюты.
- Embarked: Порт посадки (C = Cherbourg, Q = Queenstown, S = Southampton).

Жесточайший импорт библиотек!

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

Загрузка данных

```
data = pd.read_csv('tested.csv', sep=",")

data.head(10)
```

```
   PassengerId  Survived  Pclass  \
0          892         0       3
1          893         1       3
2          894         0       2
3          895         0       3
4          896         1       3
```

```
5          897         0       3
6          898         1       3
7          899         0       2
8          900         1       3
9          901         0       3

                                          Name     Sex   Age  SibSp
Parch  \
0                              Kelly, Mr. James    male  34.5      0
0
1              Wilkes, Mrs. James (Ellen Needs)  female  47.0      1
0
2                     Myles, Mr. Thomas Francis    male  62.0      0
0
3                              Wirz, Mr. Albert    male  27.0      0
0
4  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0      1
1
5                    Svensson, Mr. Johan Cervin    male  14.0      0
0
6                          Connolly, Miss. Kate  female  30.0      0
0
7                  Caldwell, Mr. Albert Francis    male  26.0      1
1
8     Abrahim, Mrs. Joseph (Sophie Halaut Easu)  female  18.0      0
0
9                        Davies, Mr. John Samuel    male  21.0      2
0

      Ticket     Fare Cabin Embarked
0     330911   7.8292   NaN        Q
1     363272   7.0000   NaN        S
2     240276   9.6875   NaN        Q
3     315154   8.6625   NaN        S
4    3101298  12.2875   NaN        S
5       7538   9.2250   NaN        S
6     330972   7.6292   NaN        Q
7     248738  29.0000   NaN        S
8       2657   7.2292   NaN        C
9  A/4 48871  24.1500   NaN        S
```

```
# Размер датасета - 418 строк, 12 колонок
data.shape
```

```
(418, 12)
```

```
# Список колонок
data.columns
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age',
'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

```
# Список колонок с типами данных
data.dtypes
```

```
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
```

Проверим на наличие пустых значений

```
data.isnull().sum()
```

```
PassengerId     0
Survived        0
Pclass          0
Name            0
Sex             0
Age            86
SibSp           0
Parch           0
Ticket          0
Fare            1
Embarked        0
dtype: int64
```

```
data = data.drop('Cabin', axis=1)
```

```
---------------------------------------------------------------------------
-----
KeyError                                  Traceback (most recent call
last)
Cell In[23], line 1
----> 1 data = data.drop('Cabin', axis=1)

File ~\AppData\Local\Packages\
```

```
PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCache\local-
packages\Python312\site-packages\pandas\core\frame.py:5581, in
DataFrame.drop(self, labels, axis, index, columns, level, inplace,
errors)
   5433 def drop(
   5434     self,
   5435     labels: IndexLabel | None = None,
   (...)
   5442     errors: IgnoreRaise = "raise",
   5443 ) -> DataFrame | None:
   5444     """
   5445     Drop specified labels from rows or columns.
   5446
   (...)
   5579             weight  1.0     0.8
   5580     """
-> 5581     return super().drop(
   5582         labels=labels,
   5583         axis=axis,
   5584         index=index,
   5585         columns=columns,
   5586         level=level,
   5587         inplace=inplace,
   5588         errors=errors,
   5589     )

File ~\AppData\Local\Packages\
PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCache\local-
packages\Python312\site-packages\pandas\core\generic.py:4788, in
NDFrame.drop(self, labels, axis, index, columns, level, inplace,
errors)
   4786 for axis, labels in axes.items():
   4787     if labels is not None:
-> 4788         obj = obj._drop_axis(labels, axis, level=level,
errors=errors)
   4790 if inplace:
   4791     self._update_inplace(obj)

File ~\AppData\Local\Packages\
PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCache\local-
packages\Python312\site-packages\pandas\core\generic.py:4830, in
NDFrame._drop_axis(self, labels, axis, level, errors, only_slice)
   4828         new_axis = axis.drop(labels, level=level,
errors=errors)
   4829     else:
-> 4830         new_axis = axis.drop(labels, errors=errors)
   4831     indexer = axis.get_indexer(new_axis)
   4833 # Case for non-unique axis
   4834 else:
```

```
File ~\AppData\Local\Packages\
PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCache\local-
packages\Python312\site-packages\pandas\core\indexes\base.py:7070, in
Index.drop(self, labels, errors)
   7068 if mask.any():
   7069     if errors != "ignore":
-> 7070         raise KeyError(f"{labels[mask].tolist()} not found in
axis")
   7071     indexer = indexer[~mask]
   7072 return self.delete(indexer)

KeyError: "['Cabin'] not found in axis"
```

```
data.isnull().sum()
```

```
PassengerId     0
Survived        0
Pclass          0
Name            0
Sex             0
Age            86
SibSp           0
Parch           0
Ticket          0
Fare            1
Embarked        0
dtype: int64
```

```
# Основные статистические характеристки набора данных
data.describe()
```

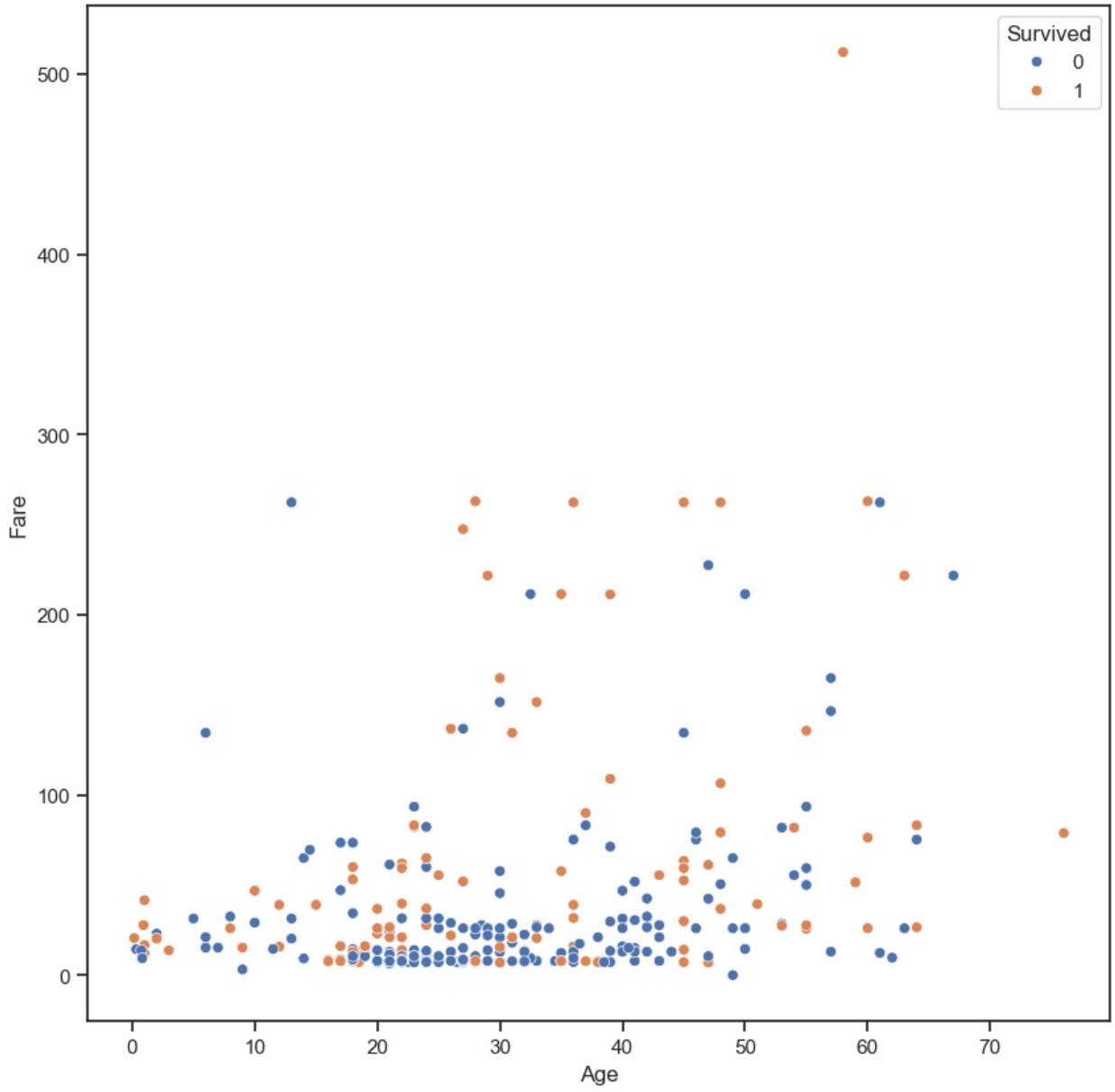
```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   418.000000  418.000000  418.000000  332.000000  418.000000
mean   1100.500000    0.363636    2.265550   30.272590    0.447368
std     120.810458    0.481622    0.841838   14.181209    0.896760
min     892.000000    0.000000    1.000000    0.170000    0.000000
25%     996.250000    0.000000    1.000000   21.000000    0.000000
50%    1100.500000    0.000000    3.000000   27.000000    0.000000
75%    1204.750000    1.000000    3.000000   39.000000    1.000000
max    1309.000000    1.000000    3.000000   76.000000    8.000000

            Parch        Fare
count  418.000000  417.000000
mean     0.392344   35.627188
std      0.981429   55.907576
min      0.000000    0.000000
25%      0.000000    7.895800
50%      0.000000   14.454200
75%      0.000000   31.500000
max      9.000000  512.329200
```

```
unique_values = data['Survived'].unique()
print(unique_values)
```

```
[0 1]
```

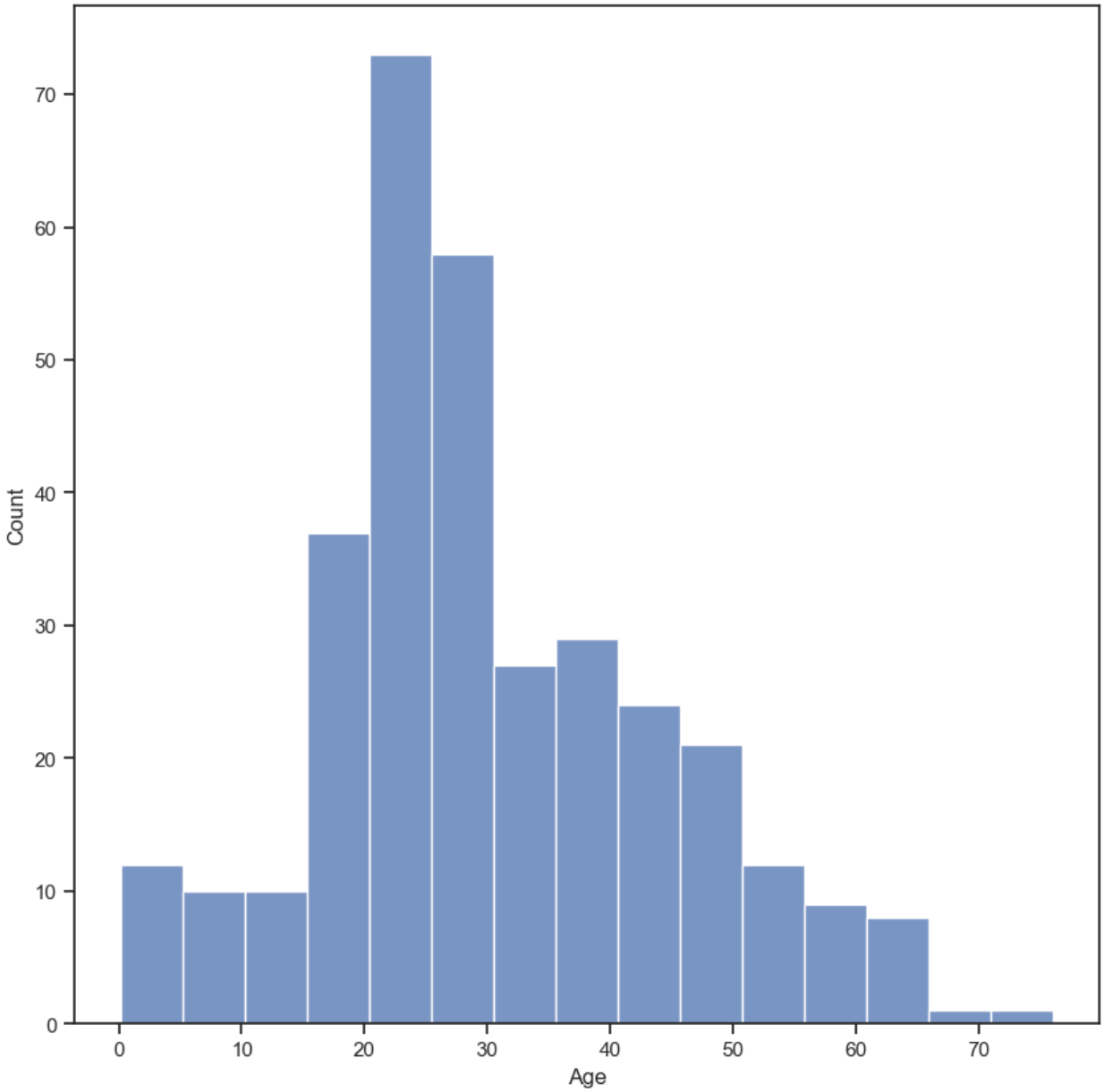
Визуальное исследование датасета

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Age', y='Fare', data=data, hue='Survived')
plt.show()
```



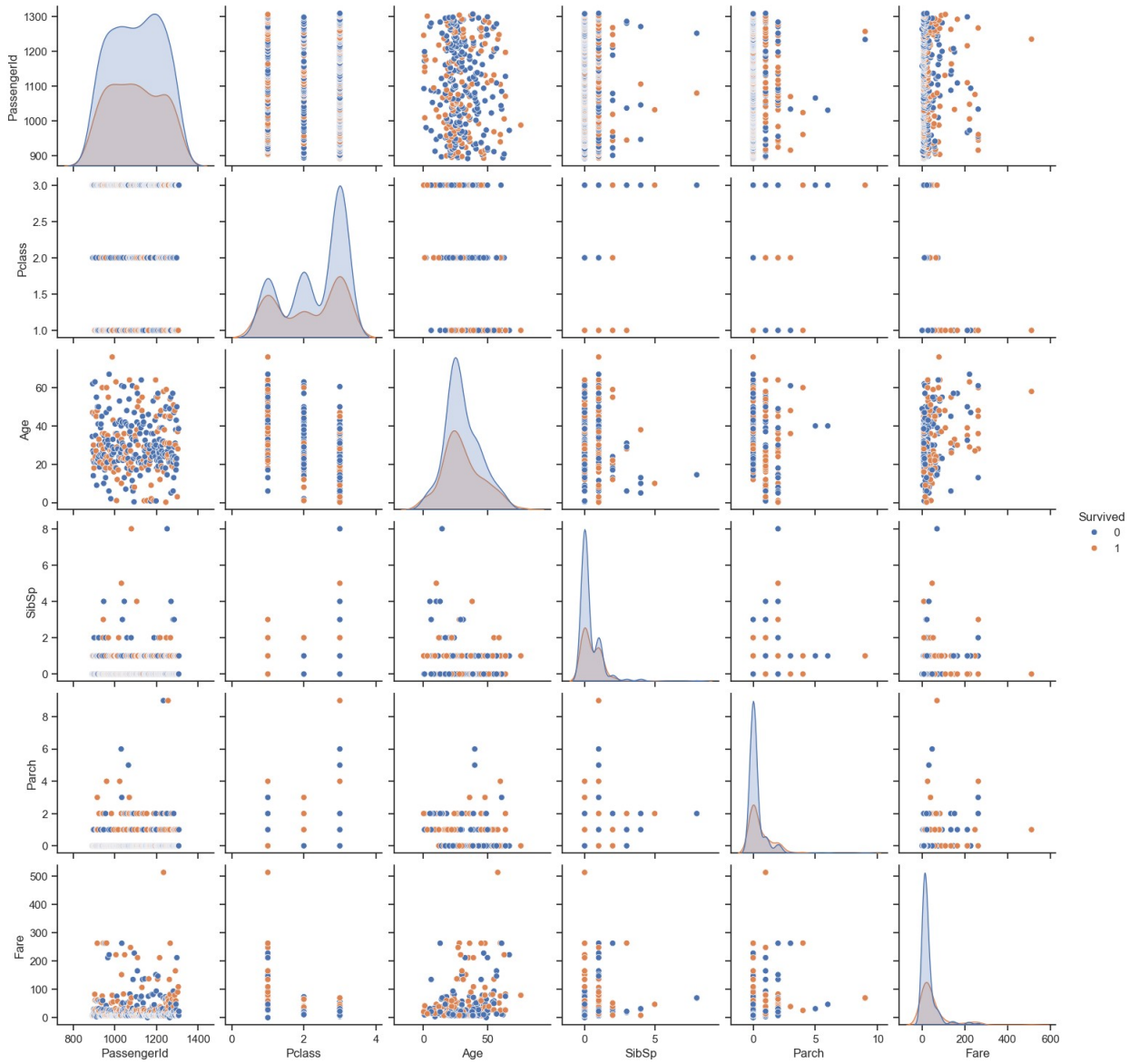
```
fig, ax = plt.subplots(figsize=(10,10))
sns.histplot(data['Age'])
```

```
<Axes: xlabel='Age', ylabel='Count'>
```



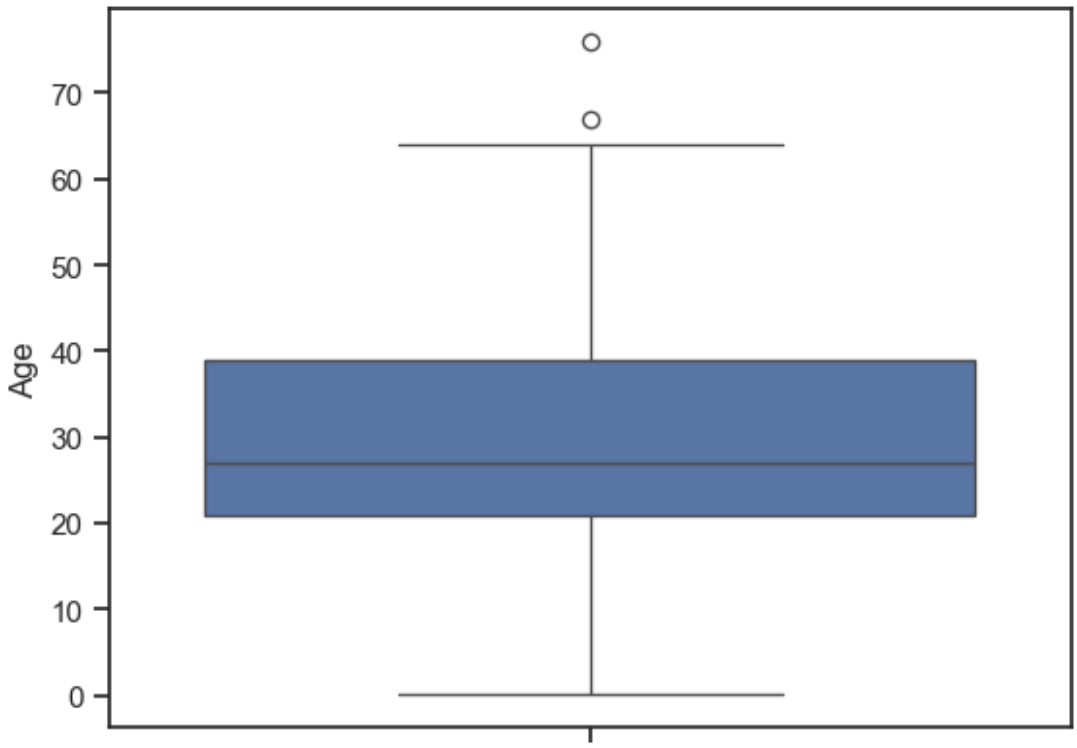

```
sns.pairplot(data, hue="Survived")
```

```
<seaborn.axisgrid.PairGrid at 0x1dbe5bc17c0>
```



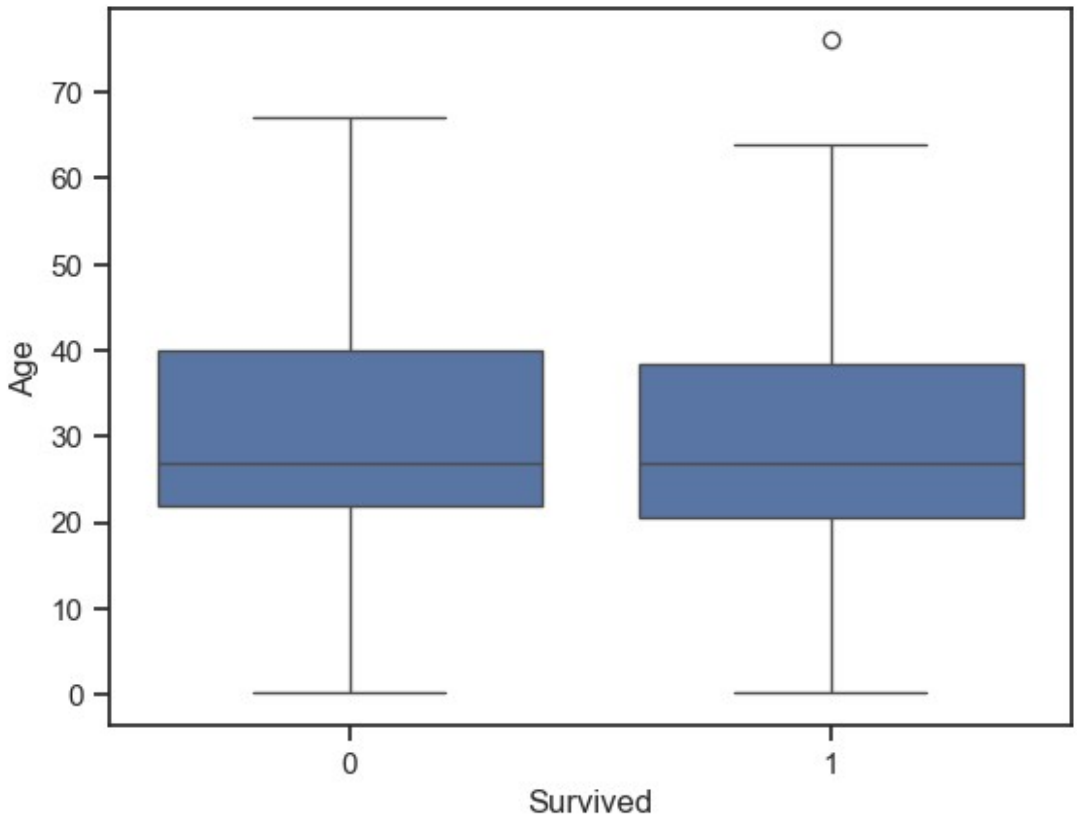

```
# По вертикали
sns.boxplot(y=data['Age'])
```

```
<Axes: ylabel='Age'>
```



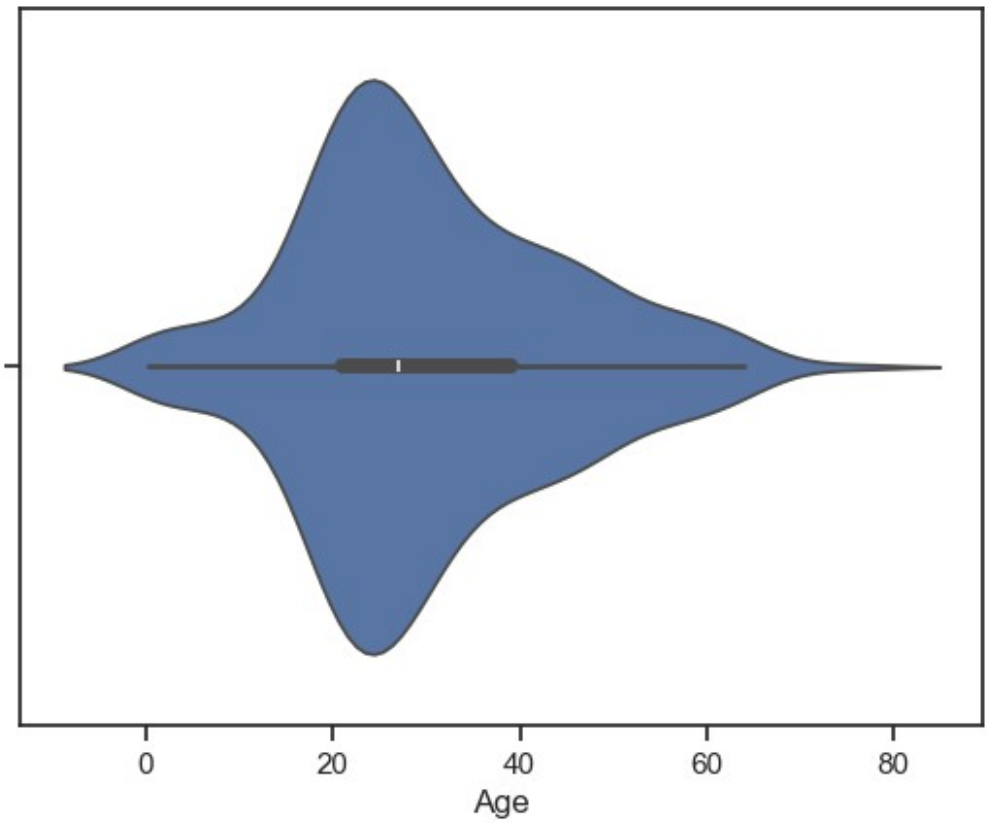

```
sns.boxplot(x='Survived', y='Age', data=data)
```

```
<Axes: xlabel='Survived', ylabel='Age'>
```



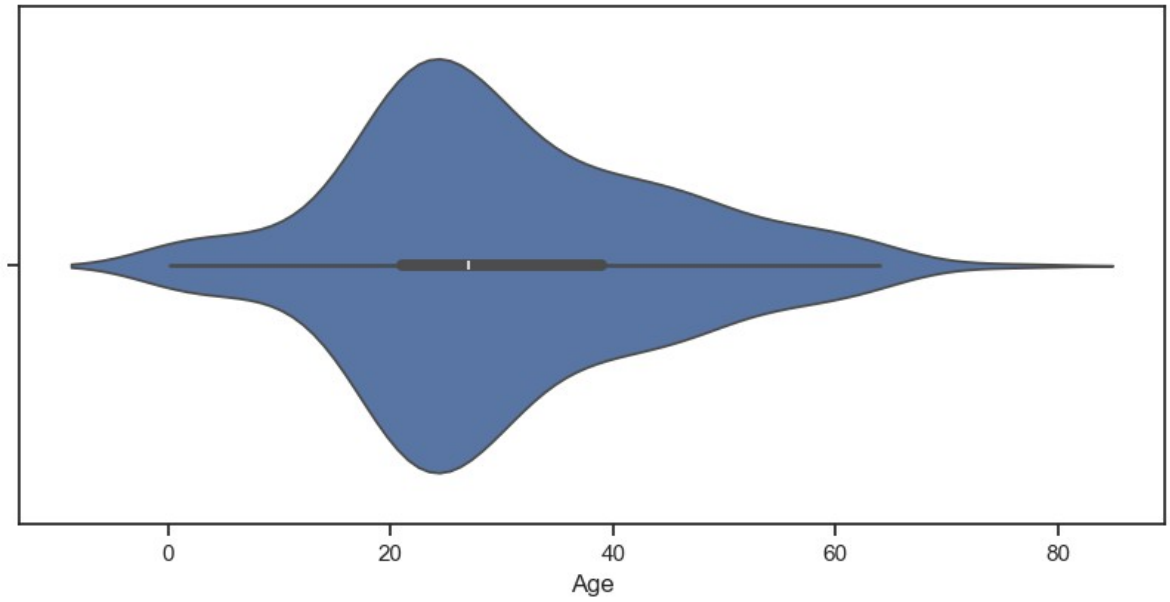

```
sns.violinplot(x=data['Age'])
```

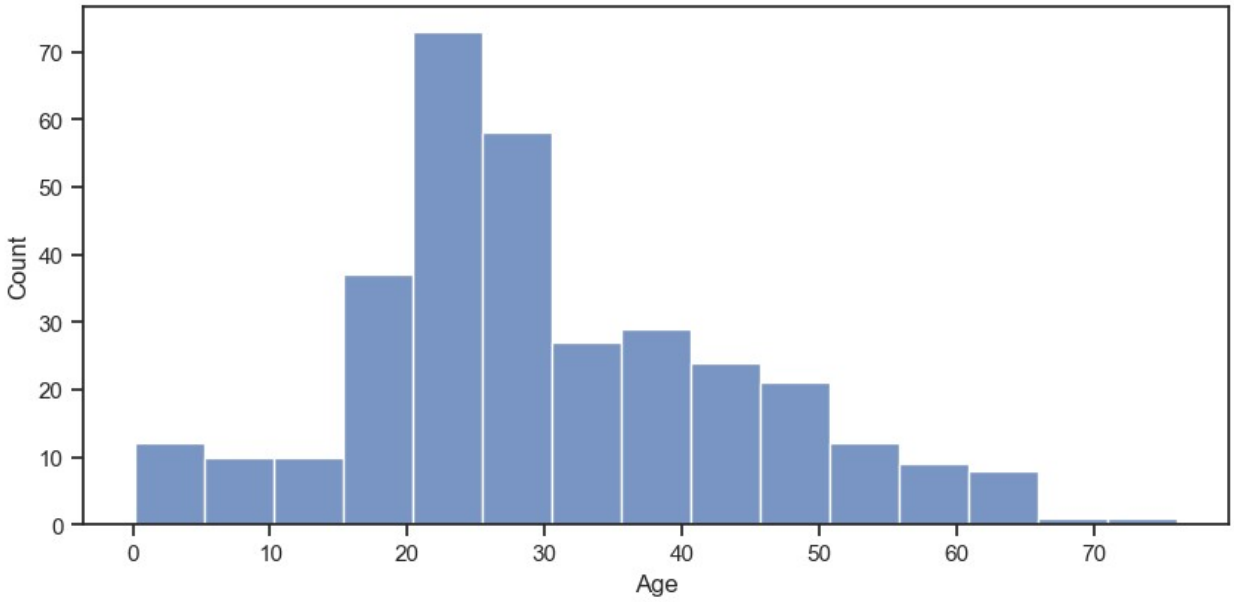
```
<Axes: xlabel='Age'>
```




```
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data['Age'])
sns.histplot(data['Age'], ax=ax[1])
```

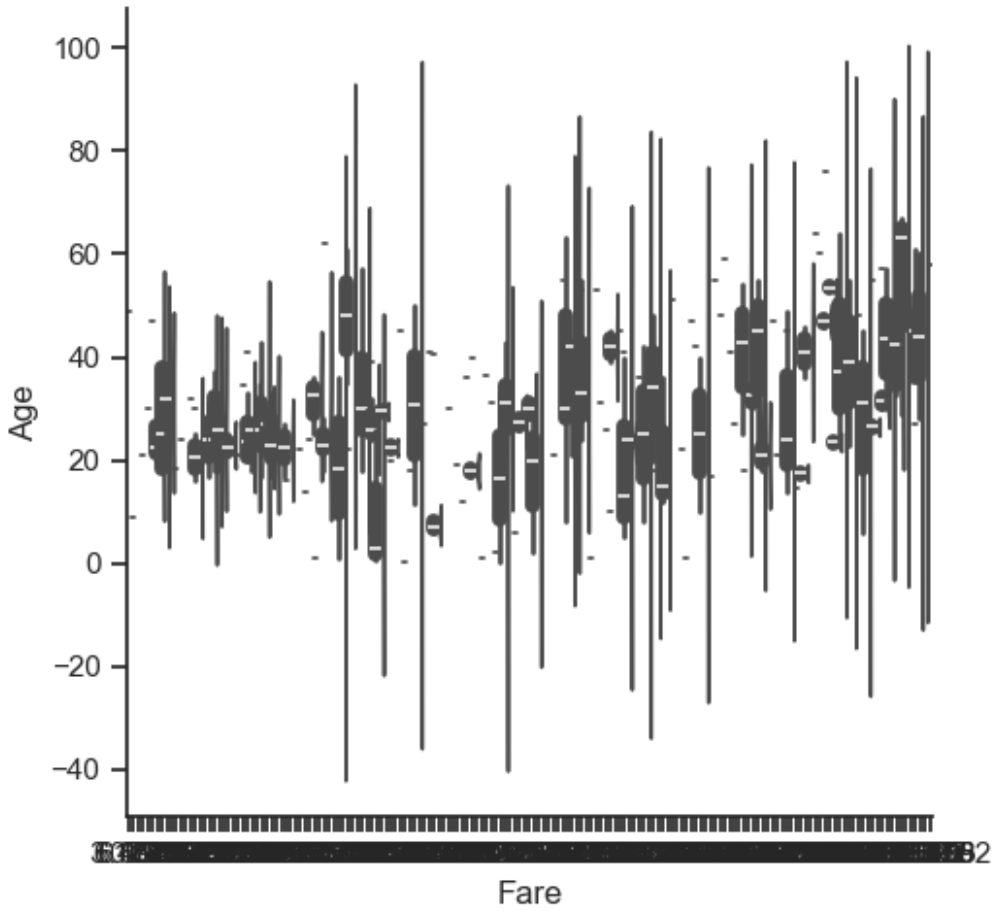
```
<Axes: xlabel='Age', ylabel='Count'>
```






```
sns.catplot(y='Age', x='Fare', data=data, kind="violin", split=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x1dbea56d7c0>
```



Информация о корреляции признаков

```
data_without_names = data.drop(columns=['Name', 'Ticket', 'Embarked'])
data_without_names['Sex'] = data_without_names['Sex'].map({'male': 0,
'female': 1})
correlation_matrix = data_without_names.corr()
print(correlation_matrix)
```

```
             PassengerId  Survived    Pclass       Sex       Age
SibSp  \
PassengerId     1.000000 -0.023245 -0.026751 -0.023245 -0.034102
0.003818
Survived       -0.023245  1.000000 -0.108615  1.000000 -0.000013
0.099943
Pclass         -0.026751 -0.108615  1.000000 -0.108615 -0.492143
0.001087
Sex            -0.023245  1.000000 -0.108615  1.000000 -0.000013
0.099943
Age            -0.034102 -0.000013 -0.492143 -0.000013  1.000000 -
0.091587
SibSp           0.003818  0.099943  0.001087  0.099943 -0.091587
1.000000
```

```
Parch            0.043080  0.159120  0.018721  0.159120 -0.061249
0.306895
Fare             0.008211  0.191514 -0.577147  0.191514  0.337932
0.171539

                 Parch      Fare
PassengerId   0.043080  0.008211
Survived      0.159120  0.191514
Pclass        0.018721 -0.577147
Sex           0.159120  0.191514
Age          -0.061249  0.337932
SibSp         0.306895  0.171539
Parch         1.000000  0.230046
Fare          0.230046  1.000000
```

Анализ корреляционной матрицы

Корреляционная матрица показывает степень взаимосвязи между различными признаками в наборе данных. Значения корреляции варьируются от -1 до 1, где:

- 1 означает полную положительную корреляцию,
- -1 означает полную отрицательную корреляцию,
- 0 означает отсутствие корреляции.

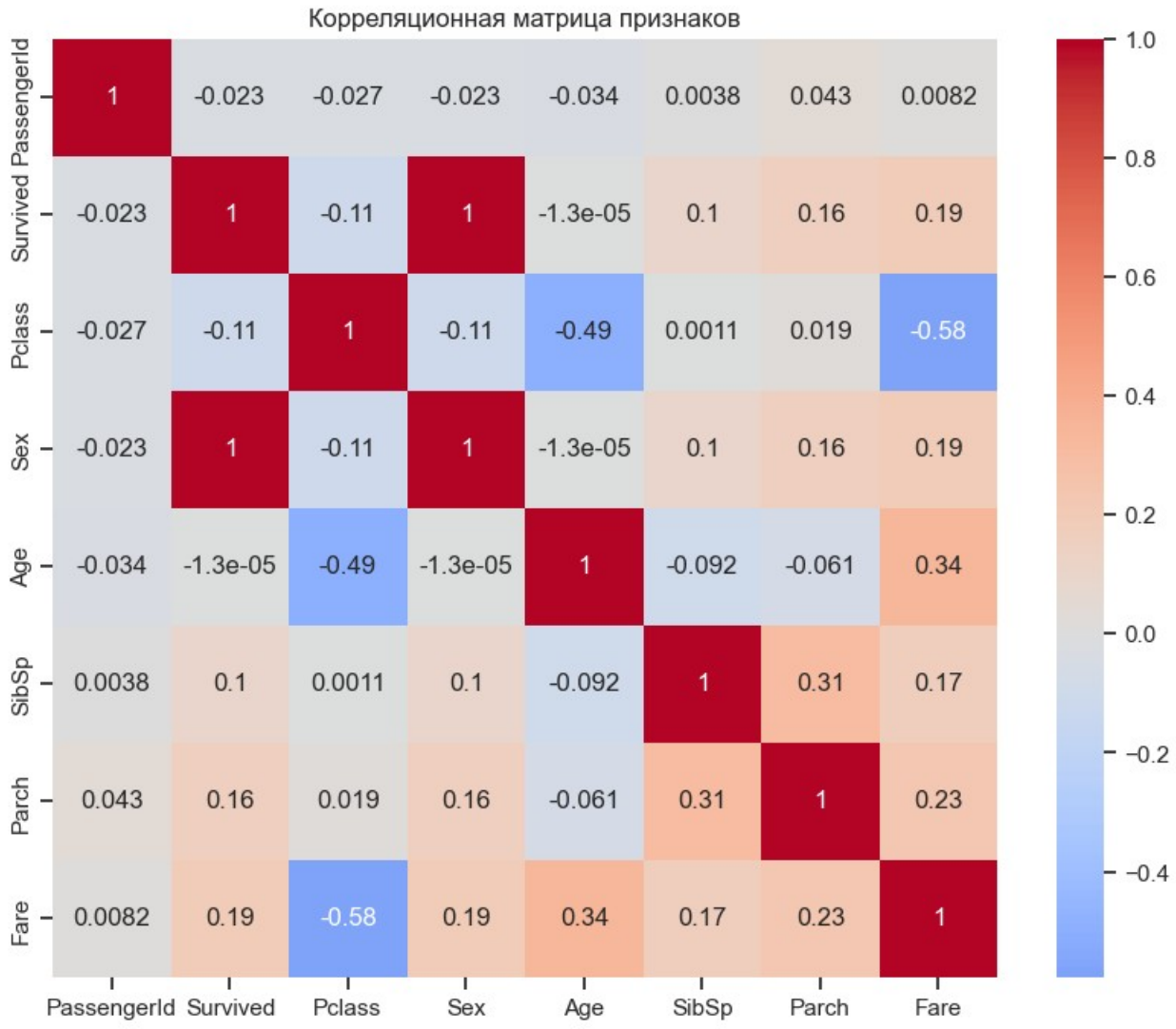
Основные наблюдения:

1. **Survived и Sex**: Значение корреляции равно 1, что указывает на полную положительную корреляцию. Это означает, что пол пассажира (мужчина или женщина) сильно влияет на вероятность выживания.
2. **Pclass и Fare**: Значение корреляции равно -0.577147, что указывает на умеренную отрицательную корреляцию. Это означает, что пассажиры с более низким классом билета (3-й класс) платили меньше за проезд.
3. **Age и Fare**: Значение корреляции равно 0.337932, что указывает на слабую положительную корреляцию. Это означает, что возраст пассажира имеет небольшое влияние на стоимость билета.
4. **SibSp и Parch**: Значение корреляции равно 0.306895, что указывает на слабую положительную корреляцию. Это означает, что количество братьев, сестер или супругов на борту связано с количеством родителей или детей на борту.

Визуализация корреляционной матрицы:

Для лучшего понимания корреляций можно построить тепловую карту (heatmap) корреляционной матрицы.

```
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0)
plt.title('Корреляционная матрица признаков')
plt.show()
```


Корреляционная матрица признаков
PassengerId
Survived
Pclass
Sex
Age
SibSp
Parch
Fare
1 -0.023 -0.027 -0.023 -0.034 0.0038 0.043 0.0082
-0.023 1 -0.11 1 -1.3e-05 0.1 0.16 0.19
-0.027 -0.11 1 -0.11 -0.49 0.0011 0.019 -0.58
-0.023 1 -0.11 1 -1.3e-05 0.1 0.16 0.19
-0.034 -1.3e-05 -0.49 -1.3e-05 1 -0.092 -0.061 0.34
0.0038 0.1 0.0011 0.1 -0.092 1 0.31 0.17
0.043 0.16 0.019 0.16 -0.061 0.31 1 0.23
0.0082 0.19 -0.58 0.19 0.34 0.17 0.23 1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4