

Database Titanic – Security information – FATEC AMERICANA

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('titanic.csv')
#Nesta Célula estamos importando os módulos pandas, matplotlib.pyplot e o seaborn,
#as funcionalidades que cada módulo fornece
```

```
In [2]: df.head()
#esse comando nos trás as 5 primeiras linhas do dataset, através dele podemos obser
#e em boa qualidade.
```

```
Out[2]:
```

	Survived	Pclass	Name	Sex	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare
0	0	3	Mr. Owen Harris Braund	male	22.0	1	0	7.2500
1	1	1	Mr . John Fadley (Fl rence Briggs Tayer) Cum...	female	38.0	1	0	71.2833
2	1	3	Miss. Laina Hei kinen	female	26.0	0	0	7.9250
3	1	1	Mrs. J cques Hea h (Lily Ma Peel) F trelle	female	35.0	1	0	53.1000
4	0	3	Mr. \ lliam Henry Allen	male	35.0	0	0	8.0500

```
In [3]: df.tail()
#o df.tail é o oposto do df.head, eles nós trás as ultimas 5 linhas do dataset, com
#que compoe o dataset
```

Out[3]:

	Survived	Pclass	Name	Sex	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare
882	0	2	Rev. Juozas Montvila	male	27.0	0	0	13.00
883	1	1	Miss. Margaret Edith Graham	female	19.0	0	0	30.00
884	0	3	Miss. Catherine Helen Johnston	female	7.0	1	2	23.45
885	1	1	Mr. Karl Howell Behr	male	26.0	0	0	30.00
886	0	3	Mr. Patrick Dooley	male	32.0	0	0	7.75

In [4]: `df.shape` #o `df.shape` é exclusivamente pra mostrar a dimensão do dataset, mostrando

Out[4]: (887, 8)

In [5]: `df.columns` #nessa célula, estamos executando o comando para ver o nome de todas as #aqui nesse caso do dataset do titanic podemos ver informações de passageiros como

Out[5]: Index(['Survived', 'Pclass', 'Name', 'Sex', 'Age', 'Siblings/Spouses Aboard', 'Parents/Children Aboard', 'Fare'], dtype='object')

In [6]: `df.dtypes` #o `df.dtypes` mostra se nas colunas existentes elas são preenchidas com alOut[6]:

Survived	int64
Pclass	int64
Name	object
Sex	object
Age	float64
Siblings/Spouses Aboard	int64
Parents/Children Aboard	int64
Fare	float64

dtype: objectIn [7]: `df.info()` #o comando `df.info` mostra a quantidade de linhas vazias nas colunas(non-n #dataset não temos nenhuma linha vazia, ou seja, com ausência de valor.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 887 entries, 0 to 886
Data columns (total 8 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Survived                             887 non-null    int64
1   Pclass                               887 non-null    int64
2   Name                                 887 non-null    object
3   Sex                                  887 non-null    object
4   Age                                  887 non-null    float64
5   Siblings/Spouses Aboard              887 non-null    int64
6   Parents/Children Aboard              887 non-null    int64
7   Fare                                 887 non-null    float64
dtypes: float64(2), int64(4), object(2)
memory usage: 55.6+ KB
```

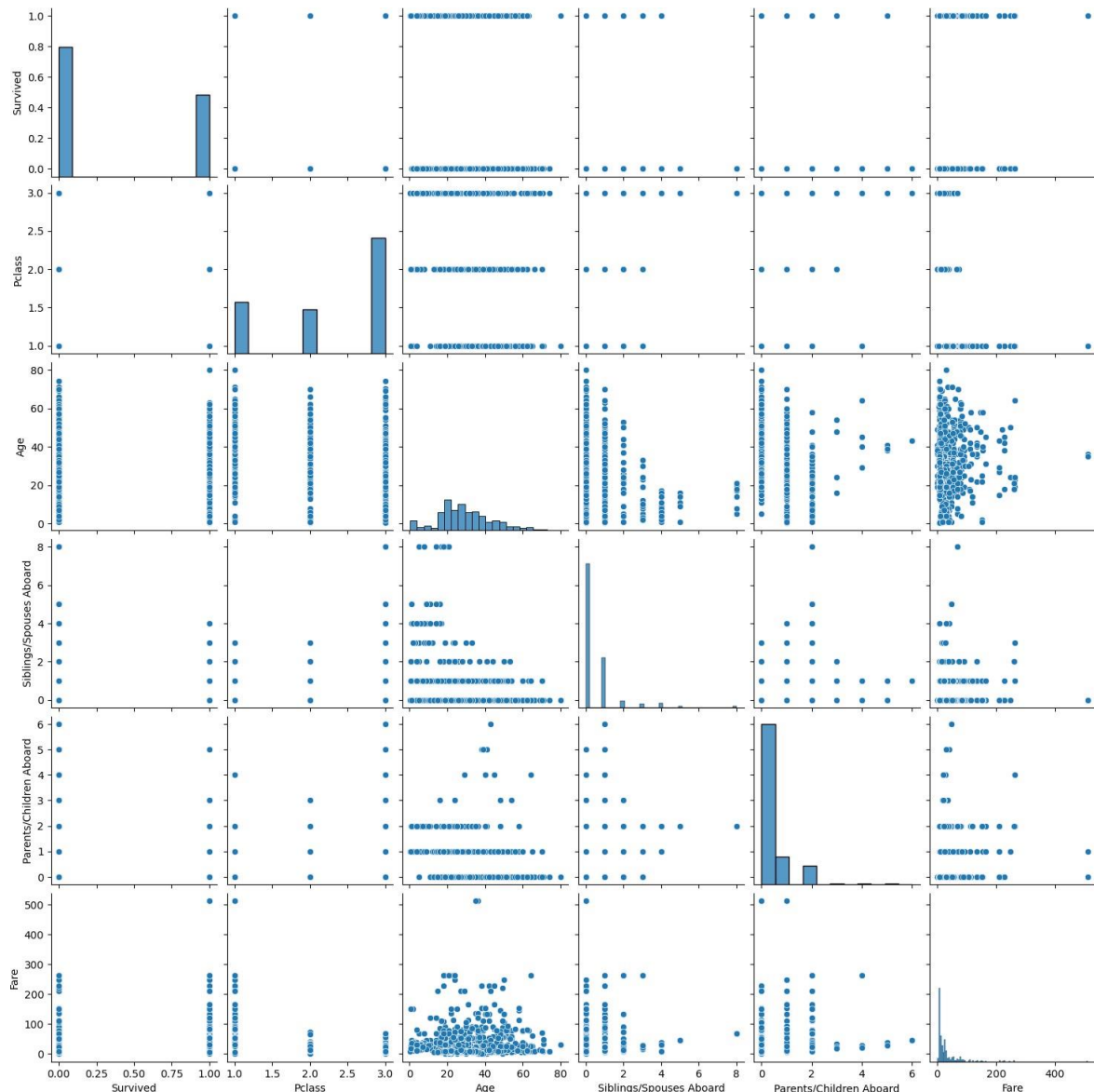
In [8]: `df.describe()`

Out[8]:

	Survived	Pclass	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare
count	887.000000	887.000000	887.000000	887.000000	887.000000	887.00000
mean	0.385569	2.305524	29.471443	0.525366	0.383315	32.30542
std	0.487004	0.836662	14.121908	1.104669	0.807466	49.78204
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.00000
25%	0.000000	2.000000	20.250000	0.000000	0.000000	7.92500
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.45420
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.13750
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.32920

In [9]:

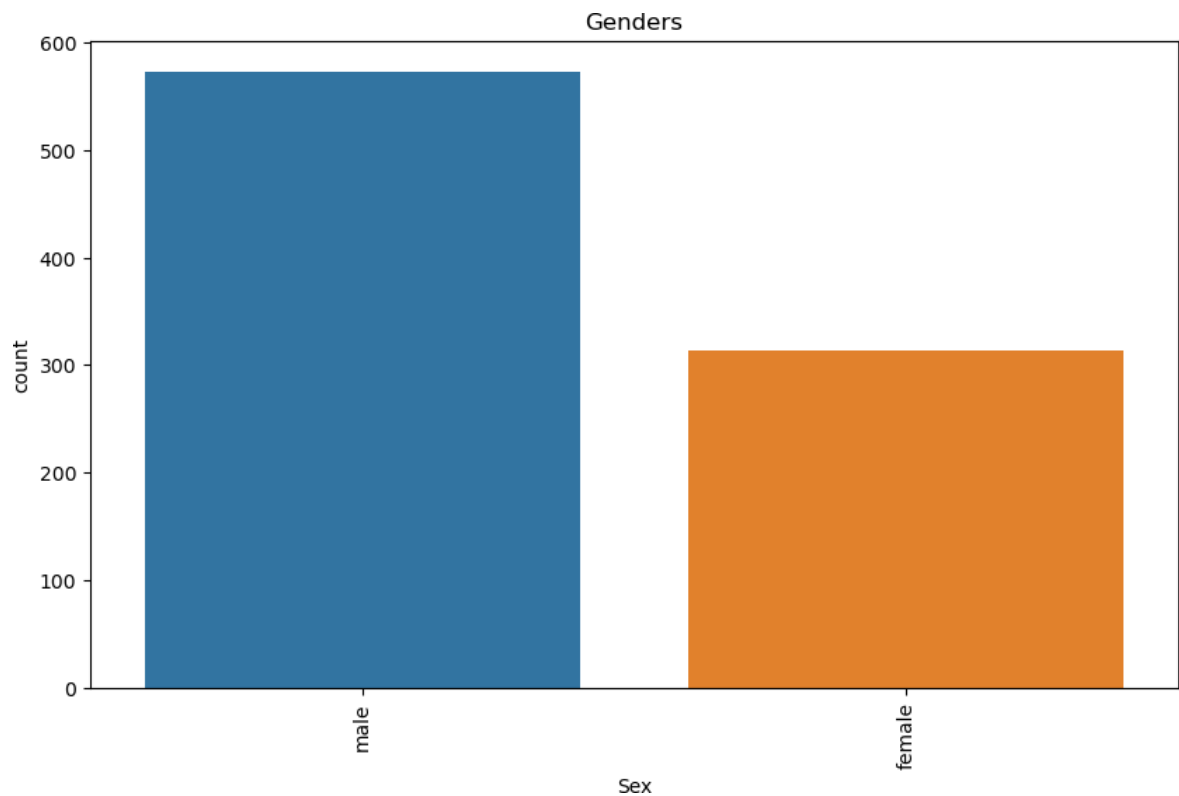
```
sns.pairplot(df)
plt.show() #com esse comando é possível criar gráficos a partir dos valores das col
```



In [10]:

```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Sex')
plt.title('Genders')
plt.xticks(rotation=90)
plt.show()
```

#Com esse código é possível criar um gráfico personalizado a partir dos valores de



In [12]: `df['Survived'].value_counts()` *#com esse comando é possível fazer a contagem dos val*

Out[12]:

0	545
1	342

Name: Survived, dtype: int64

In [24]: `df['Sex'].value_counts()` *#mesmo comando da célula acima*

Out[24]:

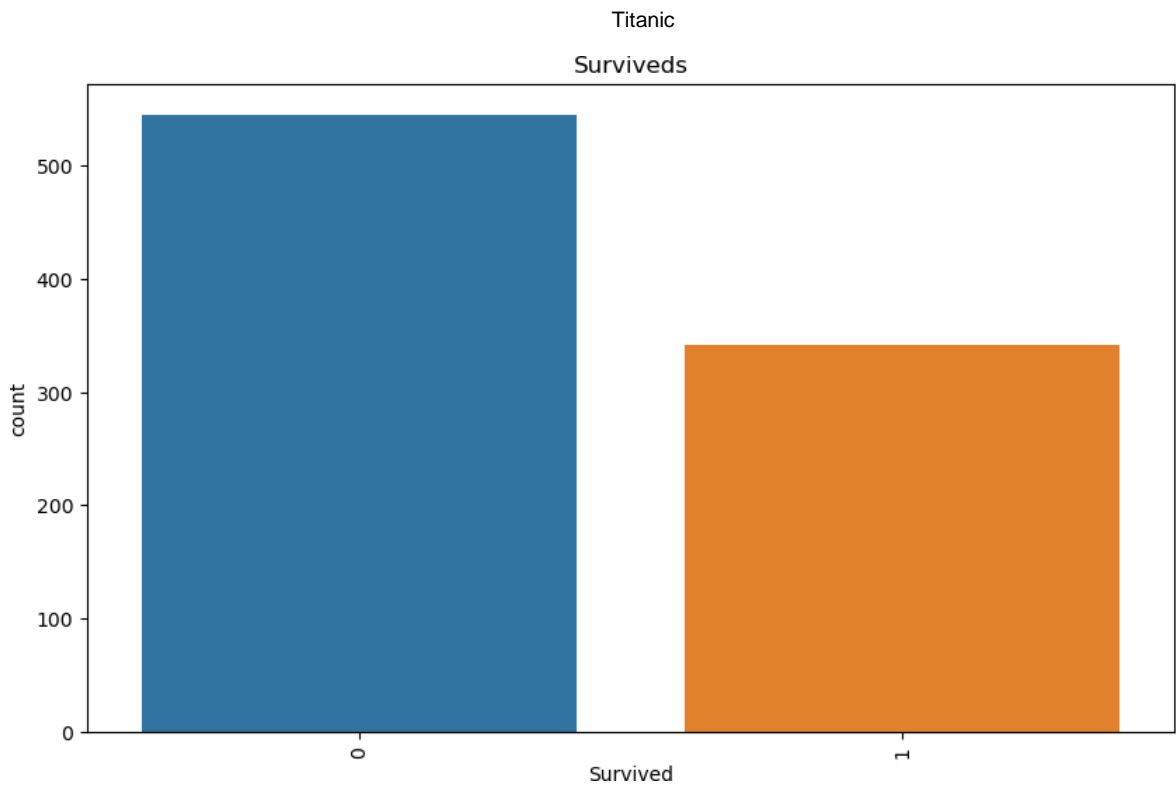
male	573
female	314

Name: Sex, dtype: int64

In [13]:

```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Survived')
plt.title('Surviveds')
plt.xticks(rotation=90)
plt.show()
```

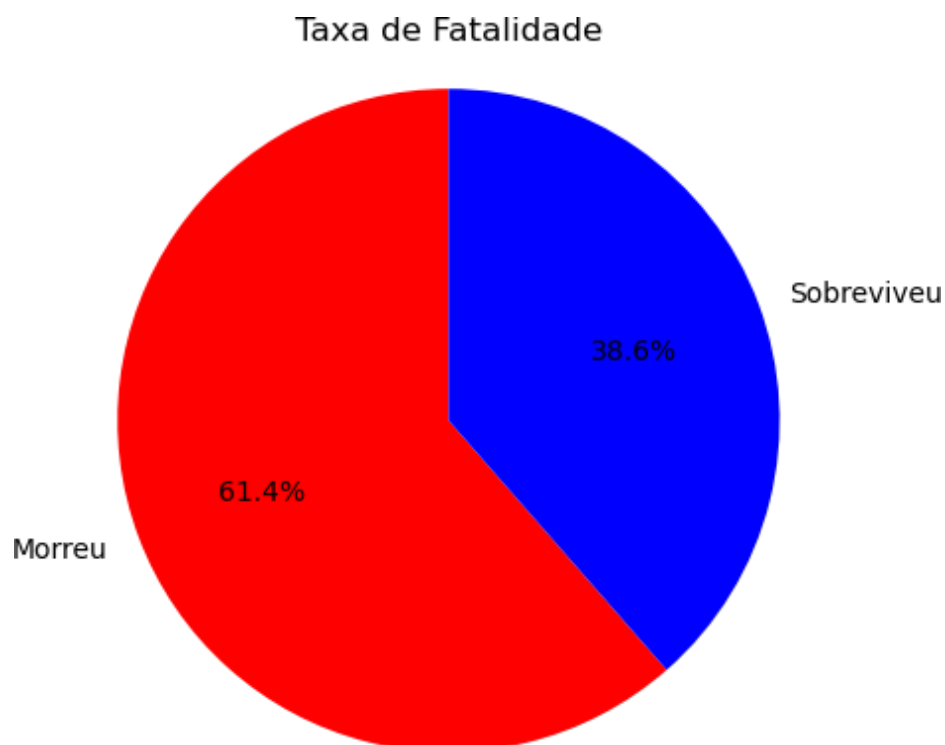
#comando do gráfico personalizado(já foi descrito acima)



```
In [23]: labels = ['Morreu', 'Sobreviveu']
        sizes = df['Survived'].value_counts()
        colors = ['red', 'blue']
        explode = (0, 0,)

        plt.pie(sizes, labels=labels, colors=colors, explode=explode, autopct='%1.1f%%', s
        plt.axis('equal')
        plt.title('Taxa de Fatalidade')
        plt.show()

        #Com esse código podemos criar um gráfico do tipo "pizza", aqui no caso deste datas
        #a taxa de fatalidade do Titanic, com a porcentagem de Mortos e Sobreviventes.
```



In []: