Exploratory Data Analysis (EDA)

```
df.shape
```

```
(891, 12)
```

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

The dataset has 12 columns and 891 rows.

Df.head() shows the top five rows of the dataset.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

This gives the data types of the columns, here we have 2 float values, 5 integer and object values each.

```
df.describe()
```

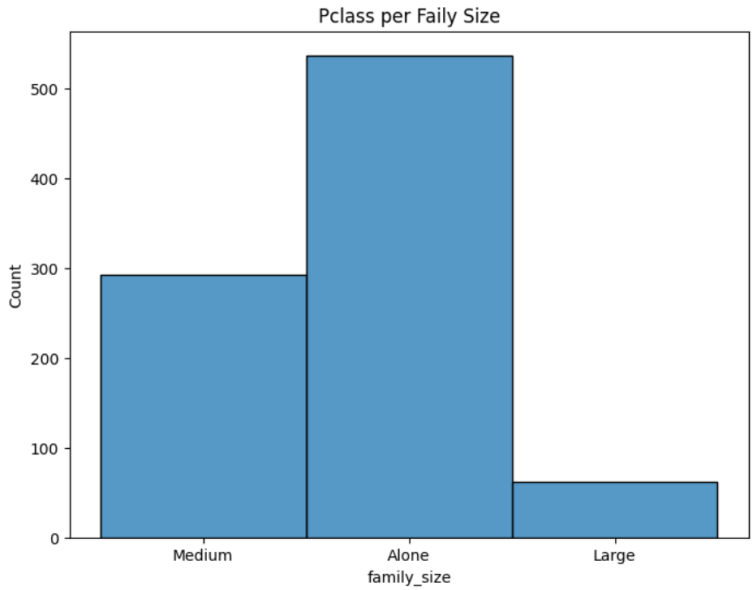
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

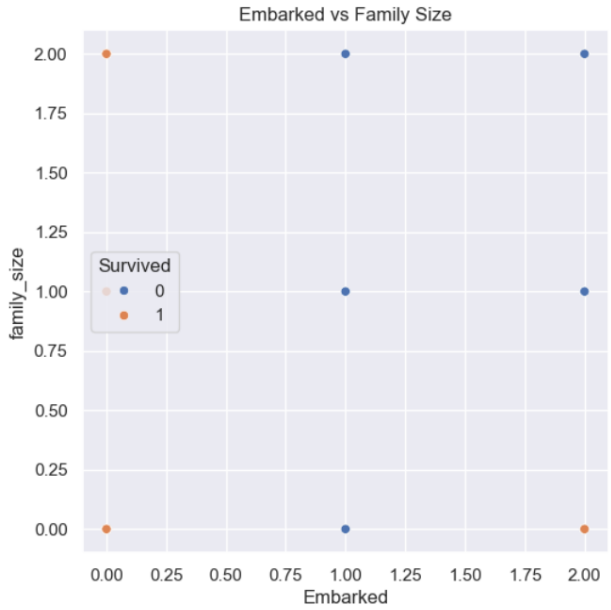
It gives the max, min, average, etc. values of all the columns. From this data we can see that there are no outliers.

```
[21]: df['family'].value_counts()
```

```
[21]: family
1     537
2     161
3     102
4      29
6      22
5      15
7      12
11      7
8       6
Name: count, dtype: int64
```

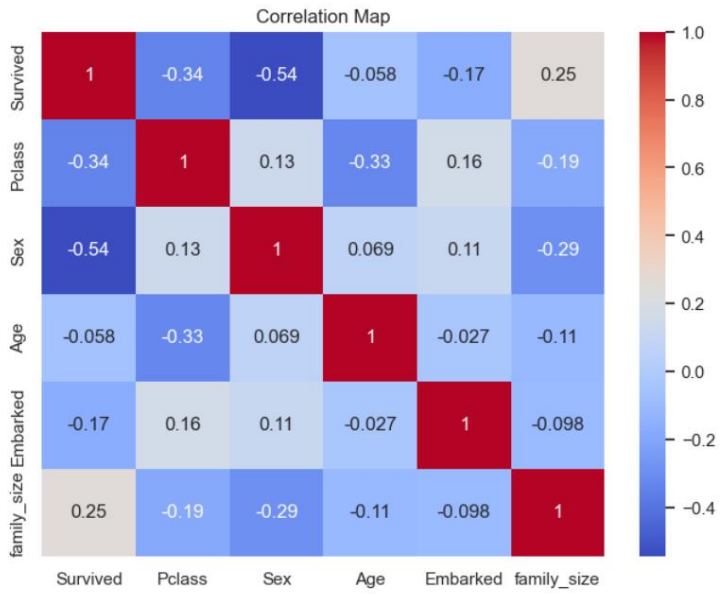
It tells us the total values per family_size





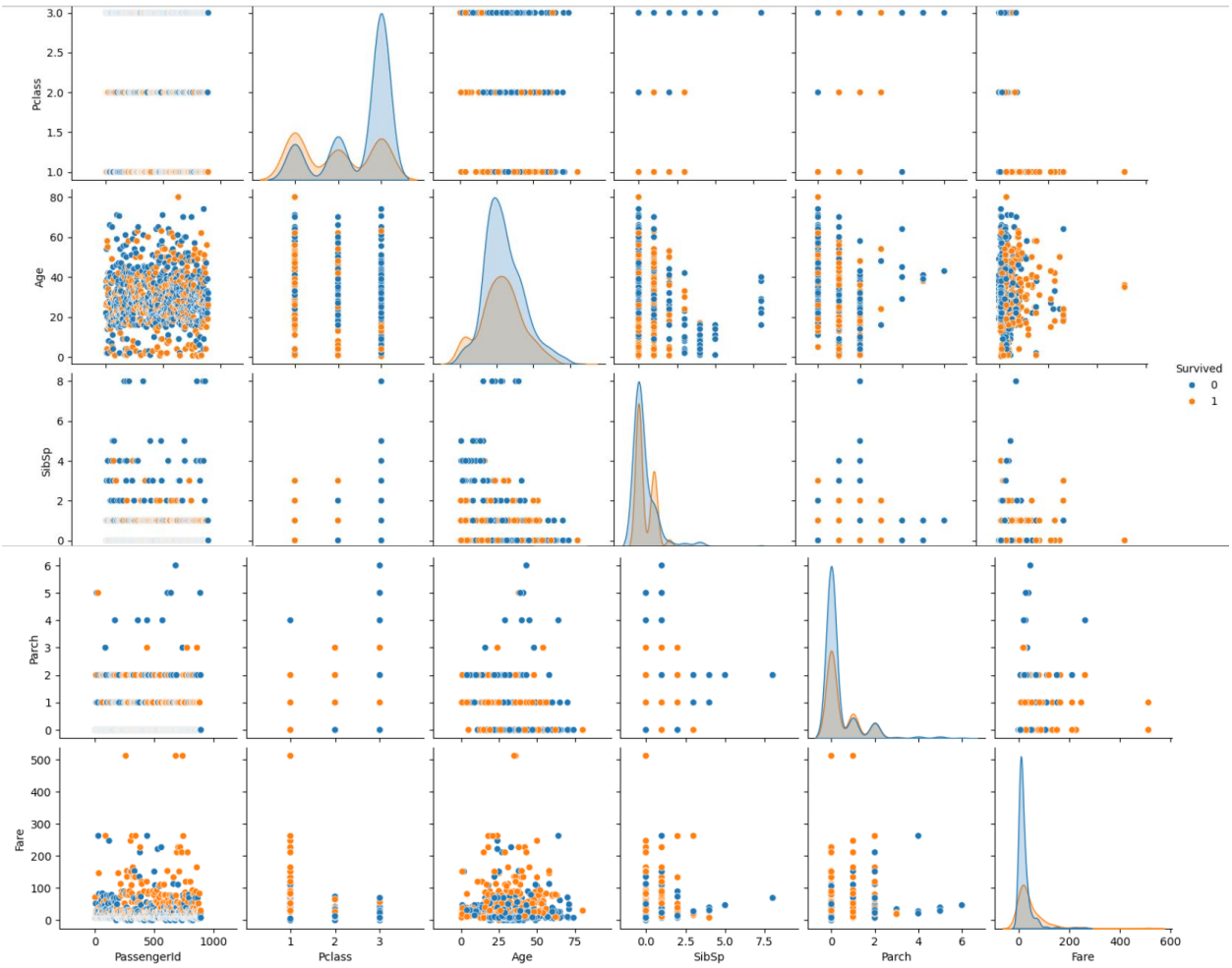
In this graph:
1 : survived, 0 : not survived

This scatter plot shows the number of survivors per family size.



This tells us the density and correlation of the column values.

Variables	Correlation Value	Meaning
Survived vs Sex	**-0.54**	Strong negative correlation → Being male (higher Sex value) **decreases** chance of survival. (Females had better survival chances.)
Survived vs Pclass	**-0.34**	Negative correlation → Higher class number (lower socio-economic status) **decreases** chance of survival. (1st class survived more.)
Survived vs Family Size	**+0.25**	Positive correlation → Having some family **slightly increased** survival chances (but not strongly).
Survived vs Embarked	**-0.17**	Weak negative correlation → Port of embarkation has a **small influence** on survival.
Survived vs Age	**-0.058**	Very weak correlation → Age alone is **not very strongly correlated** with survival.



Pclass vs Survival:

- **Higher survival** among passengers in **1st Class**.

Sex vs Survival:

- **Females** survived much more compared to **males**.

Embarked vs Survival:

- Passengers who boarded at **Cherbourg (C)** had **higher survival rates**.
- Southampton (S) had a lot of passengers, but fewer survived proportionally.

Family Size vs Survival:

- **Small** and **Medium** family sizes had **higher survival rates**.

Age vs Survival:

- **Children** (younger age groups) had better chances of survival.