just now

Task 2:- Perform data cleaning and exploratory data analysis (EDA) on a dataset of your choice, such as the Titanic dataset from Kaggle. Explore the relationships between variables and identify patterns and trends in the data.

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

```
df=pd.read_csv("C:\\Users\\Simarjeet kaur\\OneDrive\\Desktop\DS prodigy Infotech\\train.csv")
```

```
df
```

```
df.describe()
```

```
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
```

```
df.head()
```

```
df.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
```

```
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

We can see that age has 177 null values

We can Impute missing values in the "Age" column

Also there are some missing values in the "Embarked" column, we'll drop these values, and also they are relatively few.

```
df["Age"].fillna(df["Age"].median(),inplace=True)
```

```
df["Age"]
```

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888    28.0
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

```
df["Embarked"].dropna(inplace=True)
```

```
df.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

Exploring relationships between variables using correlation analysis

```
corr_mat=df.corr()
```

```
corr_mat
```

```
print(corr_mat)
```

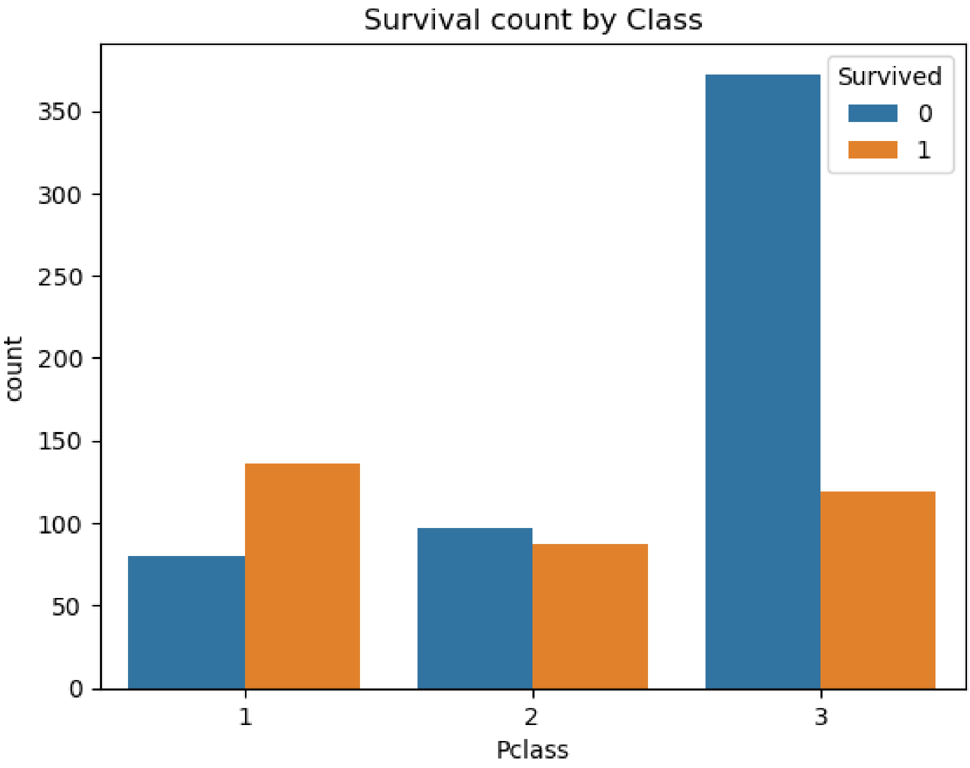
```
             PassengerId  Survived    Pclass       Age     SibSp     Parch  \
PassengerId     1.000000 -0.005007 -0.035144  0.034212 -0.057527 -0.001652
Survived       -0.005007  1.000000 -0.338481 -0.064910 -0.035322  0.081629
Pclass         -0.035144 -0.338481  1.000000 -0.339898  0.083081  0.018443
Age             0.034212 -0.064910 -0.339898  1.000000 -0.233296 -0.172482
SibSp          -0.057527 -0.035322  0.083081 -0.233296  1.000000  0.414838
Parch          -0.001652  0.081629  0.018443 -0.172482  0.414838  1.000000
Fare            0.012658  0.257307 -0.549500  0.096688  0.159651  0.216225

                 Fare
PassengerId  0.012658
Survived     0.257307
Pclass      -0.549500
Age          0.096688
SibSp        0.159651
Parch        0.216225
Fare         1.000000
```

Bar chart to visualise the distribution of survivors by class using matplotlib and seaborn libraries

```
sns.countplot(x="Pclass",hue="Survived",data=df)
plt.title("Survival count by Class")
```

```
Text(0.5, 1.0, 'Survival count by Class')
```



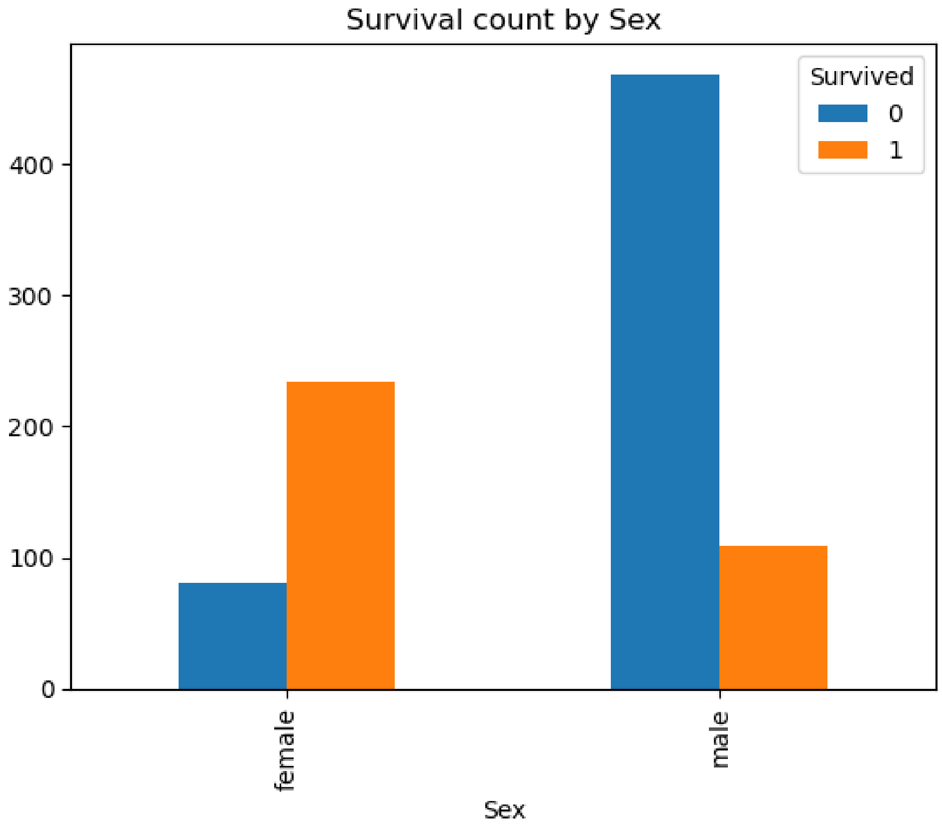
This shows us that passengers in first class had a slightly higher chance of surviving compared to those in second and third class. Most third-class passengers did not survive, while the survival rate for second-class passengers was somewhere in between.

This means that being in first class offered a better chance of survival, second class had a moderate chance, and third-class passengers were the least likely to survive.

Bar chart to visualise the distribution of survivors by sex using matplotlib and seaborn libraries

```
pd.crosstab(df["Sex"],df["Survived"]).plot(kind="bar")
plt.title("Survival count by Sex")
```

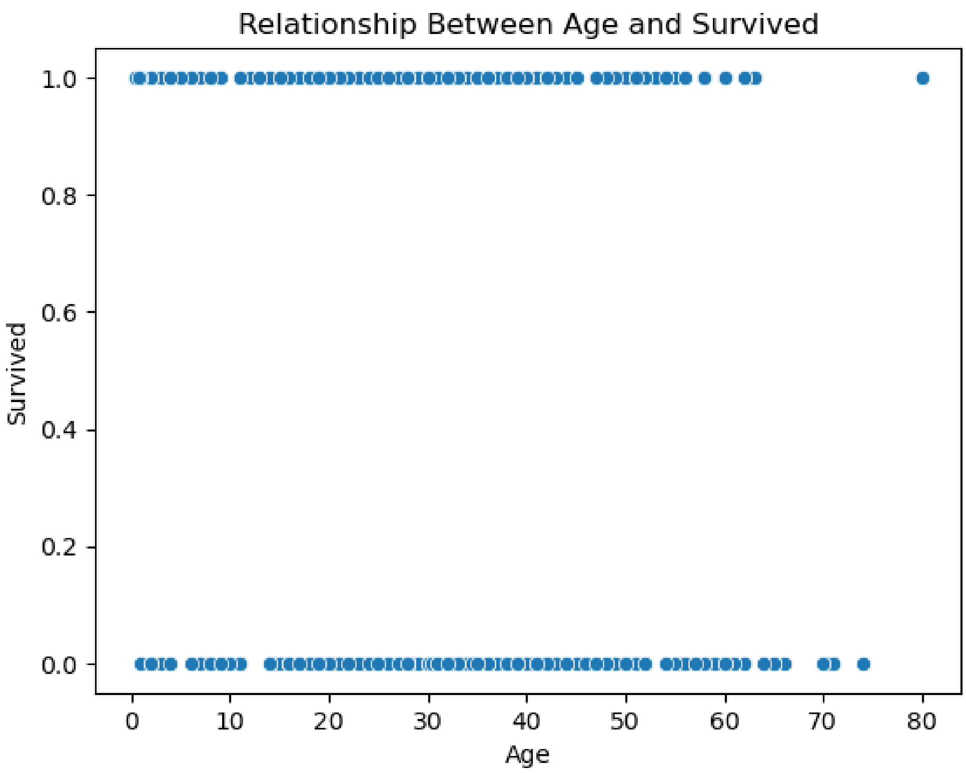
```
Text(0.5, 1.0, 'Survival count by Sex')
```



Based on the data or chart, a larger number of women survived compared to men. The pattern or trend suggests that survival rates were higher for females than for males.

Scatter Plot to visualise the relationship between "Age" and "Survived" using matplotlib and seaborn

```
sns.scatterplot(x="Age",y="Survived",data=df)
plt.title("Relationship Between Age and Survived")
plt.show()
```



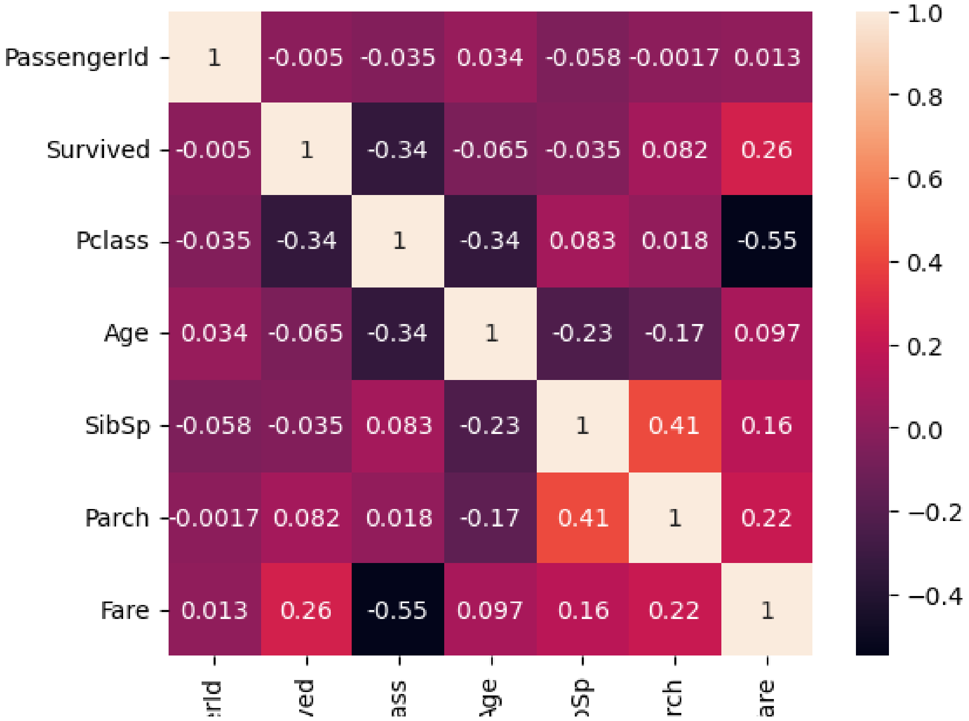
There might be some patterns regarding survival for children (especially those below 10), who seem to have a relatively higher chance of survival.

For older age groups, the plot shows survival is less predictable, and both survival and non-survival outcomes occur at various ages.

Visualizing Correlation Matrix

```
sns.heatmap(corr_mat, annot=True ,square=True)
```

```
<Axes: >
```



KEY INSIGHTS

Survived and Pclass- There is a negative correlation (-0.34) between Survived and Pclass, which means that passengers in higher class (1st Class) were most likely to survive, while those lower class had lower survival rates

Survived and Fare- There's a moderate positive correlation (0.26) between Survived and Fare, indicating that passengers who paid higher fares had a better chance of survival, which aligns with the class-based survival trend.

Pclass and Fare- A strong negative correlation (-0.55) exists between Pclass and Fare. This means that lower-class passengers paid less for their tickets, while first-class passengers paid significantly higher fares.

SibSp and Parch- There's a positive correlation (0.41) between SibSp (number of siblings/spouses aboard) and Parch (number of parents/children aboard), indicating that passengers with siblings or spouses aboard were also more likely to have parents or children aboard.

Age and Class- There's a negative correlation (-0.34) between Age and Pclass, suggesting that younger passengers were more likely to be in lower classes, while older passengers were more often in first class.

```
import jovian
```

```
jovian.commit(project="Task 2")
```