

The screenshot shows a JupyterLab environment with a notebook titled 'EDA-Task5.ipynb'. The notebook contains the following code:

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

# importing libraries and dataset

df = pd.read_csv('C:/Users/abhis/Downloads/Titanic-Dataset.csv')
```

The output of the second cell shows a statistical summary of the dataset:

```
[2]: #showing statistical summary of dataset
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

The JupyterLab interface also shows a file explorer on the left with various folders and files, and a bottom status bar indicating the current mode and file name.

The screenshot displays a JupyterLab environment. On the left, a file explorer shows a directory structure with files like Videos, Searches, Saved Games, Pictures, OneDrive, Music, Microsoft, Links, Favorites, Downloads, Documents, Desktop, Contacts, and ansel. The file EDA-Task5.ipynb is selected. The main area shows a code editor with a Python script that uses df.info() to display data types and null values for a DataFrame. The output of df.info() is visible in the code editor.

```
[3]: # shows data types and null values
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
```

The image shows a JupyterLab interface. On the left is a file explorer pane showing a directory structure with files like Videos, Searches, Saved Games, Pictures, OneDrive, Music, Microsoft, Links, Favorites, Downloads, Documents, Desktop, Contacts, and ansel. The file 'EDA-Task5.ipynb' is selected. The main area is a code editor for 'EDA-Task5.ipynb' in 'Python 3 (ipykernel)' mode. The code editor shows a series of cells. The first cell displays data types for 'Cabin' and 'Embarked' columns. The second cell shows the command `df['Survived'].value_counts()` and its output, which is a series with two categories: 0 (549) and 1 (342). The third cell shows the command `df['Pclass'].value_counts()` and its output, which is a series with three categories: 3 (491), 1 (216), and 2 (184). The fourth cell shows the command `df['Sex'].value_counts()` and its output, which is a series with two categories: male (577) and female (314). The fifth cell shows the command `df['Embarked'].value_counts()` and its output, which is a series with three categories: S (644), C (168), and Q (77). The bottom status bar shows 'Simple' mode, 'Python 3 (ipykernel) | Idle', and the current position 'Ln 1, Col 30'.

EDA-Task5.ipynb - JupyterLab

localhost:8888/lab/tree/EDA-Task5.ipynb

Suggested Sites Maps YouTube Web Store Translate Gmail Yellow line flame tra... Nightbot - Dashboa... Dashboard - YouTube Dashboard / Stream... Chat Box Widget Disney+ Hotstar ... YouTube Music Other favorites

File Edit View Run Kernel Tabs Settings Help

+

EDA-Task5.ipynb

now

EDA-Task5.ipynb

Notebook Python 3 (ipykernel)

[11]: #identifying null values  
df.isnull().sum()

[11]: 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

[13]: # handling null values  
df['Age'] = df['Age'].fillna(df['Age'].median())

[14]: df['Cabin\_known'] = df['Cabin'].notna().astype(int)  
df = df.drop(columns=['Cabin'])

[15]: df['Embarked'] = df['Embarked'].fillna(df['Embarked'].mode()[0])

[16]: df.isnull().sum()

[16]: PassengerId 0  
Survived 0  
Pclass 0  
Name 0  
Sex 0  
Age 0  
SibSp 0  
Parch 0  
Ticket 0  
Fare 0  
Embarked 0  
Cabin\_known 0  
dtype: int64

[ ]:

Observation:  
-Missing values in Age, Cabin, and Embarked.  
-Cabin is mostly missing

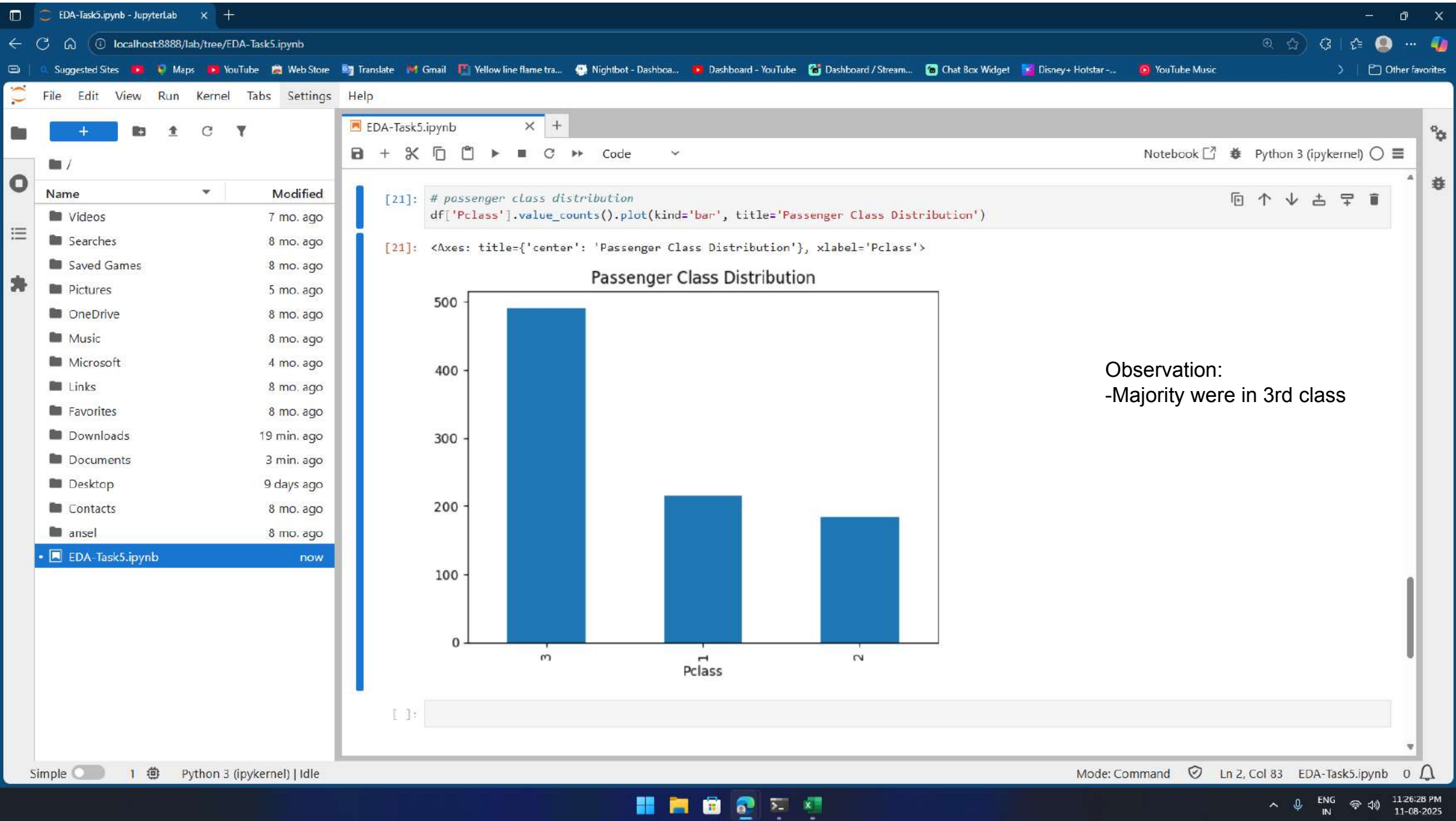
Simple 1 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 18 EDA-Task5.ipynb 0 11:19:57 PM 11-08-2025

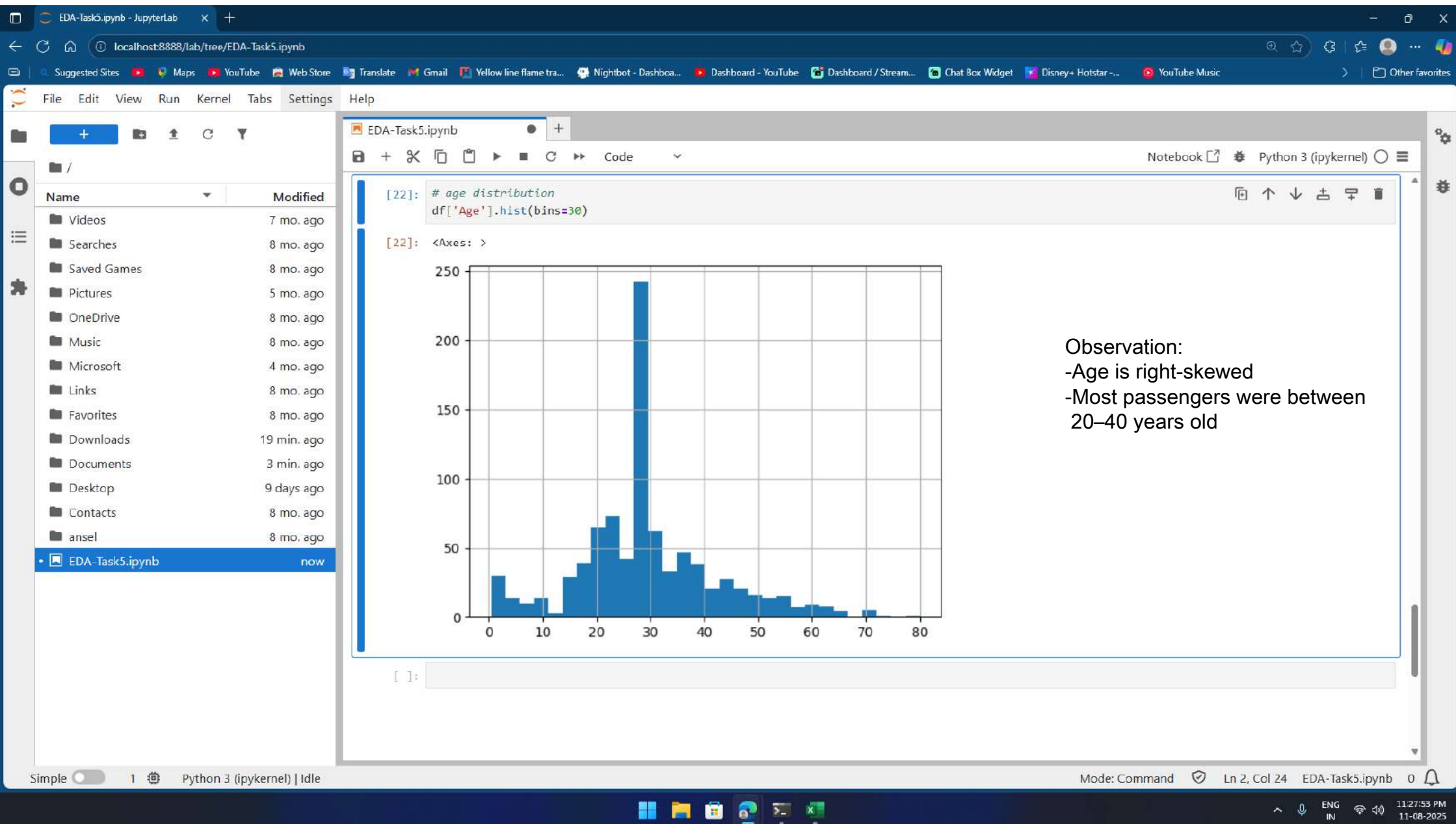


The screenshot displays a JupyterLab environment with a file explorer on the left and a code editor on the right. The file explorer shows a list of files and folders, with 'EDA-Task5.ipynb' selected. The code editor shows two lines of Python code that generate a bar chart. The first line is a comment: `# survival distribution`. The second line is `df['Survived'].value_counts().plot(kind='bar', title='Survival Count')`. The output of the code is a bar chart titled 'Survival Count'. The x-axis is labeled 'Survived' and has two categories: '0' and '1'. The y-axis represents the count, ranging from 0 to 500. The bar for '0' is approximately 550 units high, and the bar for '1' is approximately 340 units high. To the right of the chart, there is a text observation: 'Observation: -38% of passengers survived'. The JupyterLab interface includes a top menu bar with 'File', 'Edit', 'View', 'Run', 'Kernel', 'Tabs', and 'Help'. The bottom status bar shows 'Simple', 'Python 3 (ipykernel)', 'Mode: Command', 'Ln 2, Col 71', 'EDA-Task5.ipynb', and the time '11:25:50 PM 11-08-2025'.

Survived	Count
0	550
1	340

Observation:  
-38% of passengers survived





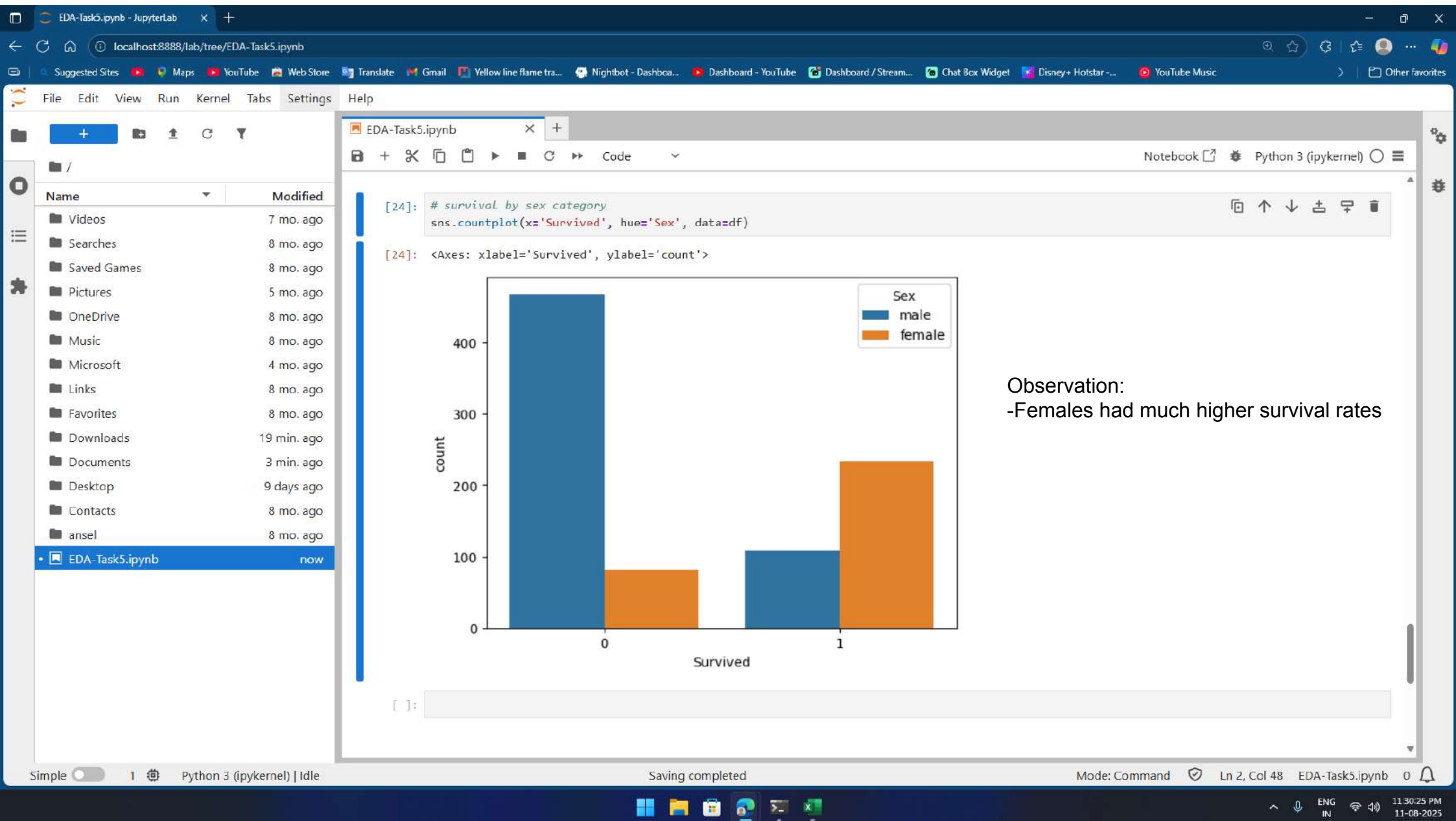
The screenshot displays a JupyterLab environment with a file browser on the left and a code editor on the right. The file browser shows a directory structure with files like 'Videos', 'Searches', 'Saved Games', 'Pictures', 'OneDrive', 'Music', 'Microsoft', 'Links', 'Favorites', 'Downloads', 'Documents', 'Desktop', 'Contacts', and 'ansel'. The code editor shows a Python script that plots the fare distribution using a box plot. The plot shows a median fare around 20, with a box from approximately 10 to 40. There are several outliers, with the highest one reaching over 500. The text 'Observation: -Fare has extreme outliers -Median fare is low' is written next to the plot.

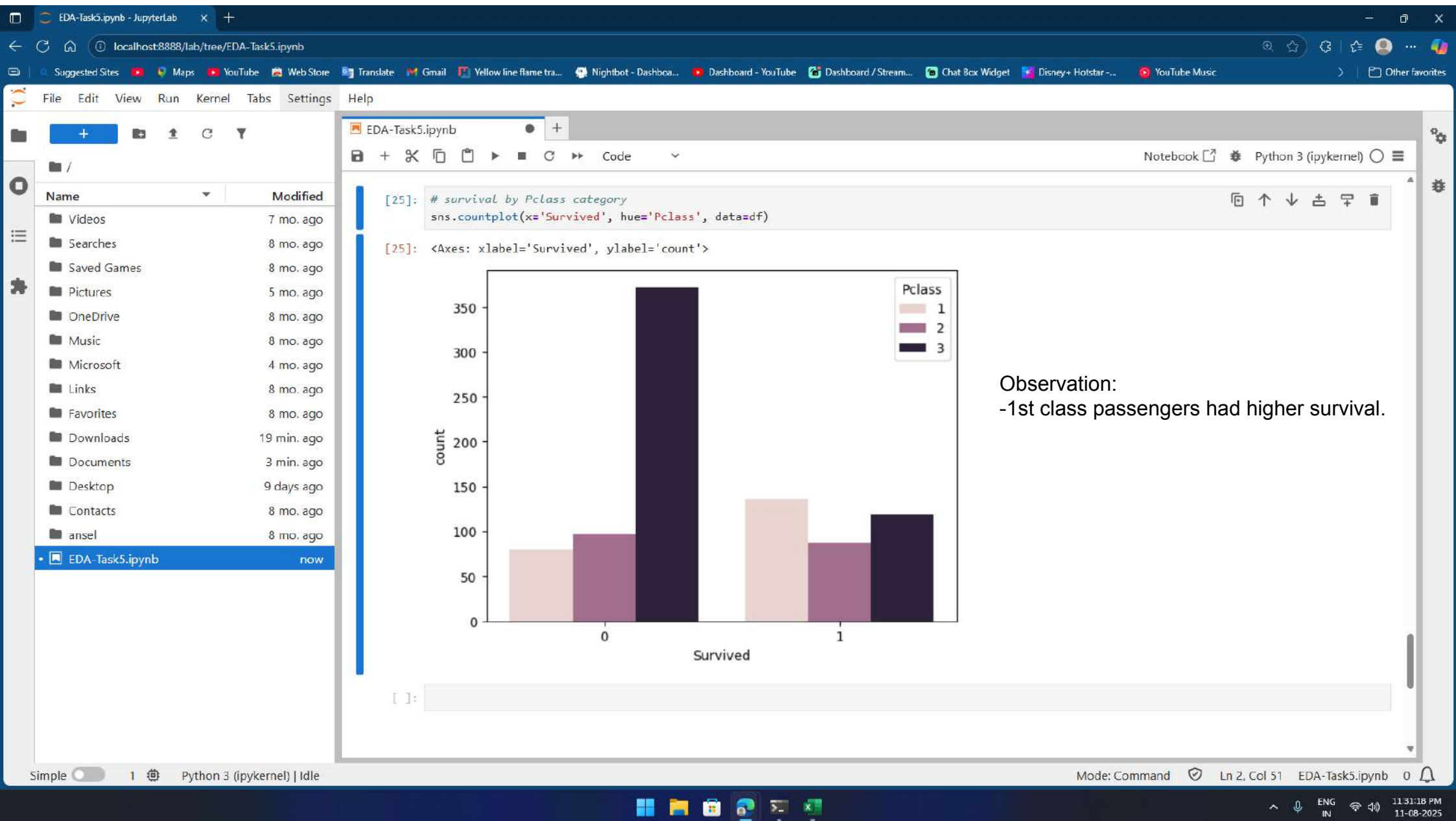
```
[23]: # fare distribution
df['Fare'].plot(kind='box')

[23]: <Axes: >
```

Observation:  
-Fare has extreme outliers  
-Median fare is low







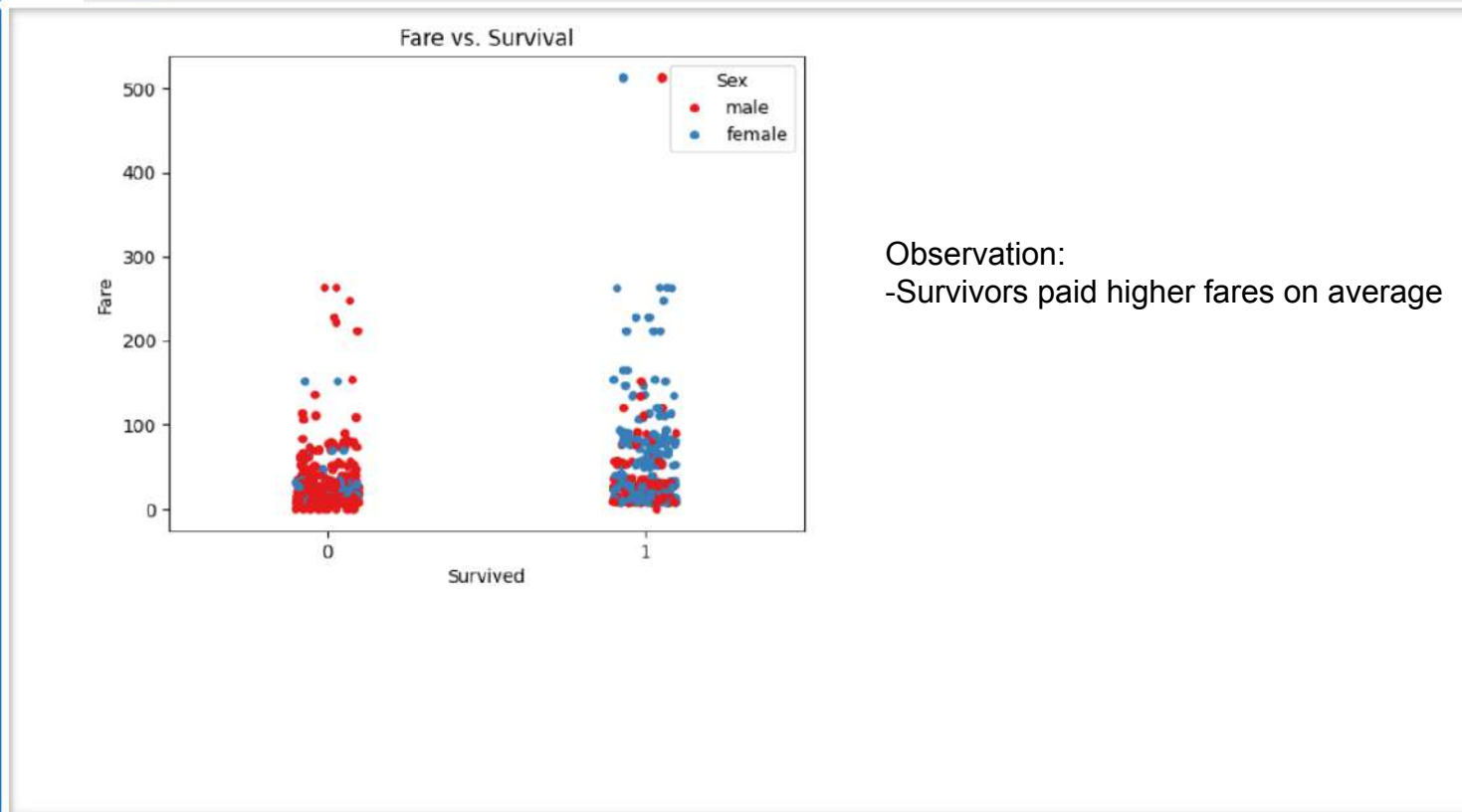
The screenshot displays a JupyterLab interface with a file browser on the left and a notebook editor on the right. The file browser shows a list of files and folders, with 'EDA-Task5.ipynb' selected. The notebook editor shows the following code and output:

```
[26]: # age vs survival
sns.boxplot(x='Survived', y='Age', data=df)
```

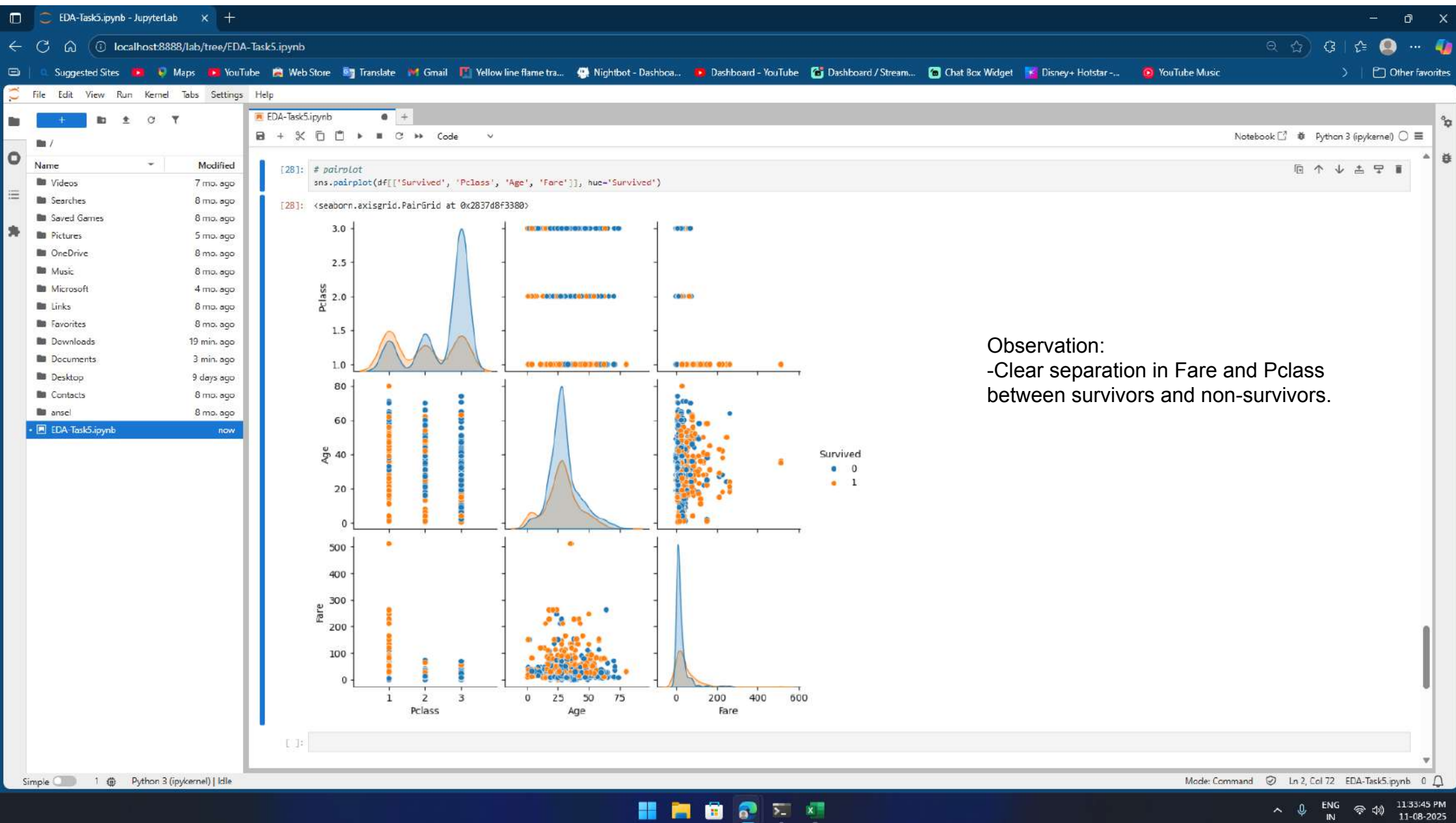
The output is a boxplot titled 'Age vs Survived'. The x-axis is labeled 'Survived' with categories 0 and 1. The y-axis is labeled 'Age' with a range from 0 to 80. The boxplot shows that survivors (Survived=1) have a slightly higher median age (around 28) compared to non-survivors (Survived=0, around 25). There are many outliers for both groups, particularly for non-survivors.

Observation:  
-Survivors tended to be slightly younger

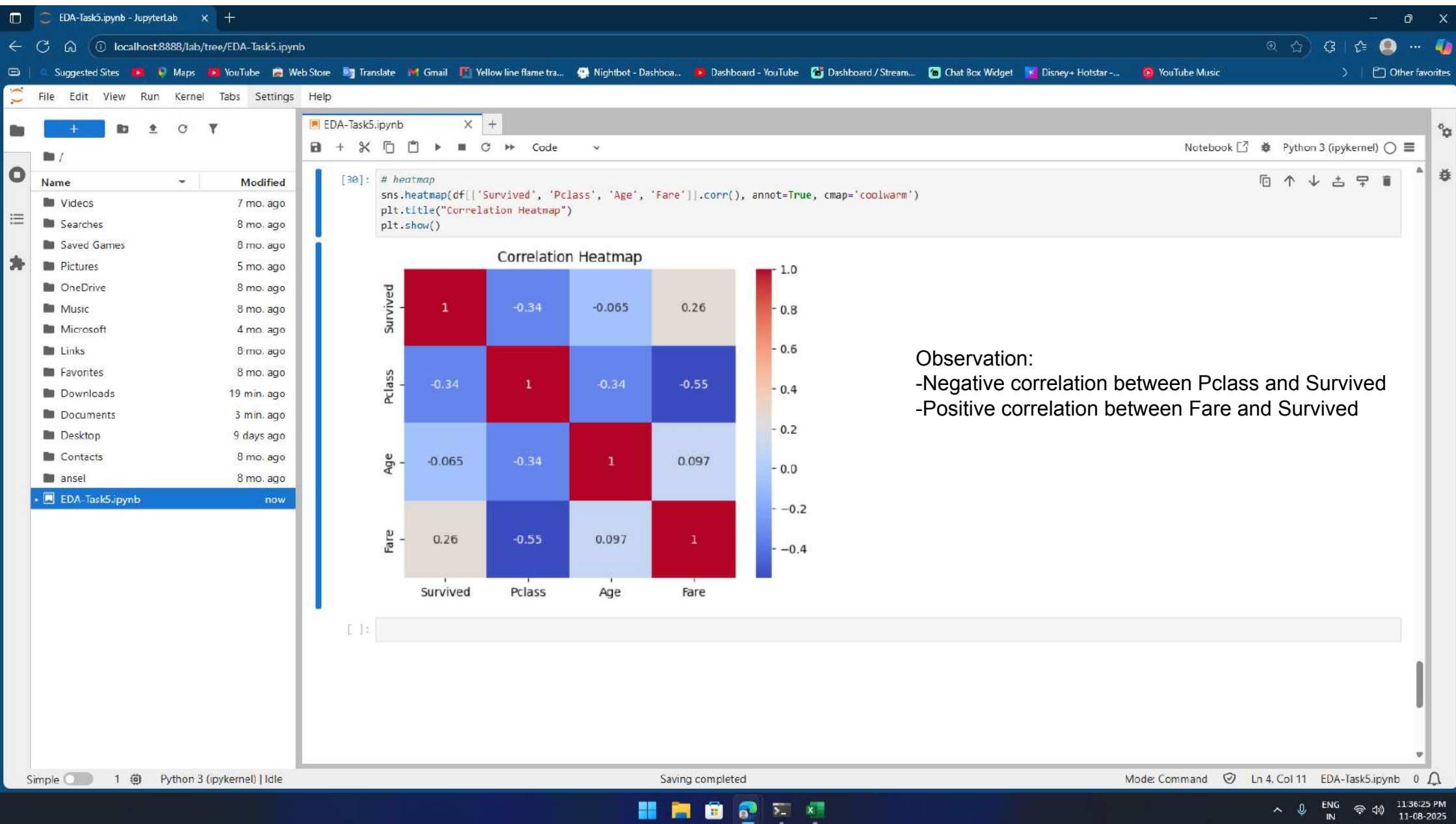
```
[8]: # fare vs survival
sns.stripplot(x='Survived', y='Fare', hue='Sex', data=df, jitter=True, palette='Set1')
plt.title("Fare vs. Survival")
plt.show()
```



Observation:  
-Survivors paid higher fares on average







# Summary of Findings:

- Survival Rate: 38% of passengers survived.
- Gender: Females had significantly higher survival rates than males.
- Passenger Class: 1st class passengers had the highest survival rate whereas 3rd class had the lowest.
- Age: Younger passenger had slightly better survival, though survival spanned all age groups.
- Fare: Higher fares were associated with better survival odds, likely due to class and cabin quality.
- Embarked: Most passengers embarked from Southampton; missing values were minimal and filled with mode.
- Cabin: Cabin data was mostly missing, a binary feature (Cabin\_known) was created to retain some signal.
- Missing Data: `Cabin` mostly missing; `Age` and `Embarked` need cleaning. `Age` was imputed with median, `Embarked` with mode whereas `Cabin` was dropped after feature engineering.
- Correlations: Fare and Passenger class showed moderate correlation with survival, Age had weak correlation.