DATA SCIENCE BOOTCAMP

Naufal Raif Al-Zaky

LIBRARY

```
import matplotlib.pyplot as plt
import seaborn as sns

import pandas as pd

# Load dataset
df = pd.read_csv('estonia-passenger-list.csv')
df
```

	PassengerId	Country	Firstname	Lastname	Sex	Age	Category	Survived
0	1	Sweden	ARVID KALLE	AADLI	M	62	P	0
1	2	Estonia	LEA	AALISTE	F	22	C	0
2	3	Estonia	AIRI	AAVASTE	F	21	C	0
3	4	Sweden	JURI	AAVIK	M	53	C	0
4	5	Sweden	BRITTA ELISABET	AHLSTROM	F	55	P	0
...	...	...	...	...	...	...	...	...
984	985	Sweden	ANNA INGRID BIRGITTA	OSTROM	F	60	P	0
985	986	Sweden	ELMAR MIKAEL	OUN	M	34	P	1
986	987	Sweden	ENN	QUNAPUU	M	77	P	0
987	988	Sweden	LY	GUNAPUU	F	87	P	0
988	989	Sweden	CARL	OVBERG	M	42	P	1

989 rows × 8 columns

Import 3 libraries
matplotlib: Create images or graphs such as bar and line charts
seaborn: Create neater and more pleasing graphs.
pandas: to read data files and process tabular data

The dataset is taken from the estonia-passenger-list.csv file. The data is loaded using pandas, and displayed as a table. The data size (989, 8) means there are 989 rows and 8 columns. The columns include PassengerId, Country, Firstname, Lastname, Sex, Age, Category, and Survived.

MISSING VALUES

```
# Memeriksa jumlah nilai yang hilang (missing values) dalam setiap kolom.
print(df.isnull().sum())
```

```
PassengerId    0
Country        0
Firstname      0
Lastname       0
Sex            0
Age            0
Category       0
Survived       0
dtype: int64
```

The code **print(df.isnull().sum())** is used to check the number of missing values in each column. The results shown show that all columns (PassengerId, Country, Firstname, Lastname, Sex, Age, Category, and Survived) have a value of 0, meaning there is no missing data in this dataset.

```
# Mengisi nilai yang hilang pada kolom Age dengan median usia.
df['Age'].fillna(df['Age'].median(), inplace=True)
# Mengisi nilai yang hilang pada kolom 'Survived' dengan modus (nilai yang paling sering muncul).
df['Survived'].fillna(df['Survived'].mode()[0], inplace=True)
```

The code is used to handle empty data in the dataset. Empty values in the Age column are filled with the median age, while empty values in the Survived column are filled with the mode (the most frequently occurring value).

DATAFRAME

```
[ ]  # Memeriksa jumlah baris duplikat dalam DataFrame
     print(f"Jumlah duplikat: {df.duplicated().sum()}")
```

```
Jumlah duplikat: 0
```

This code is used to check the number of duplicate rows in a DataFrame (df). The df.duplicated() function returns a boolean value for each row (True if the row is duplicate), then .sum() counts the number of True (i.e. the number of duplicate rows).

```
[5]  # Menghapus baris duplikat dari DataFrame.
     df.drop_duplicates(inplace=True)
```

This code is used to remove duplicate rows from DataFrame df. The **drop_duplicates(inplace=True)** method will directly remove duplicate rows inside the df object without having to store them into a new variable.

STATISTICS

```
# Menghasilkan statistik deskriptif untuk kolom numerik dalam DataFrame
print(df.describe())
```

```
       PassengerId         Age    Survived
count   989.000000  989.000000  989.000000
mean    494.992922   44.575329    0.138524
std     285.643660   17.235146    0.345624
min       1.000000    0.000000    0.000000
25%     248.000000   30.000000    0.000000
50%     495.000000   44.000000    0.000000
75%     742.000000   59.000000    0.000000
max     989.000000   87.000000    1.000000
```

df.describe() provides summary statistics for the numeric columns in the DataFrame, namely: PassengerId, Age, and Survived.

PASSENGERID

- count (number of data): 989 rows
- mean (average): 494.99 average passenger ID
- std (standard deviation): 285.64 shows an even distribution of IDs
- min - max: from 1 to 989 shows that each passenger has a unique ID

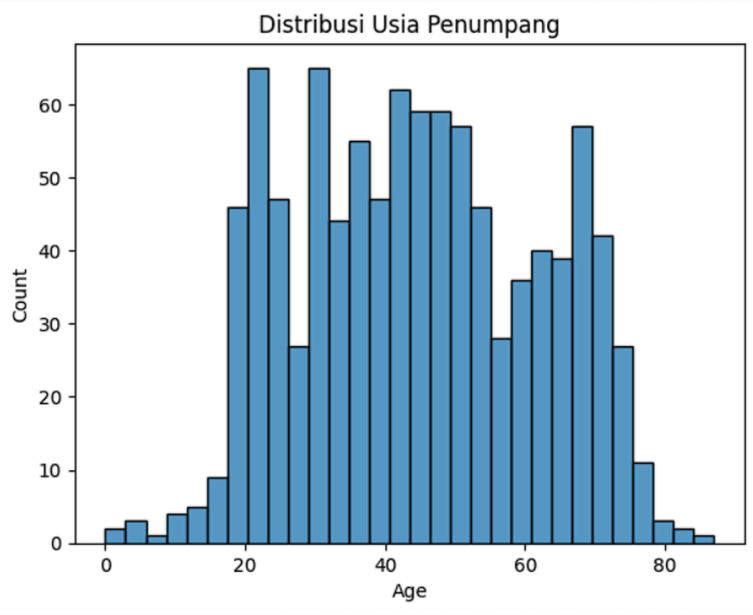
AGE

- count: 989 no age data missing
- mean: 44.58 years average passenger age
- std: 17.24 quite varied age
- min - max: 0 – 87 years there are babies to the elderly

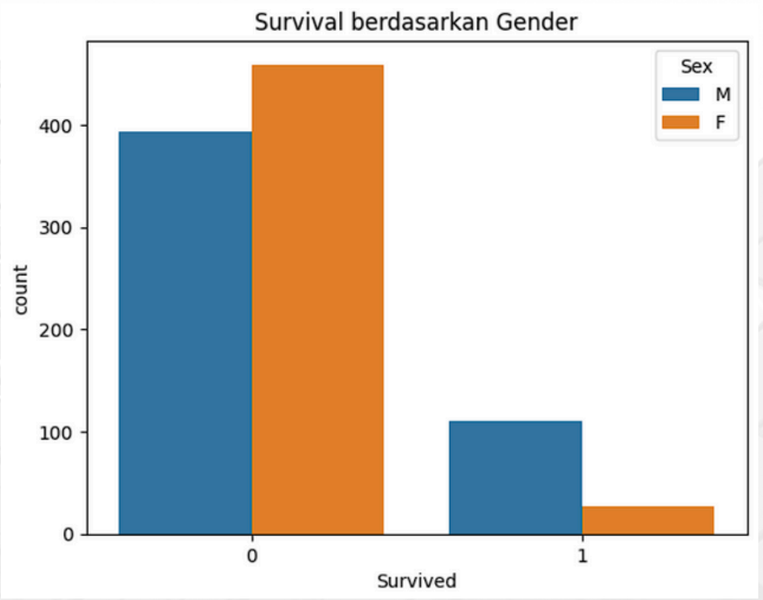
SURVIVED

- count: 989 no empty data
- mean: 0.138 only 13.8% of passengers survived
- std: 0.35 binary value (0 = not survived, 1 = survived)
- min - max: 0 – 1 only two values, because this is a binary category column

VISUALIZATION



The first visualization uses a histogram (sns.histplot) to show the age distribution of passengers. This graph divides the age range into 30 parts (bins) and shows how many passengers are in each age group.



The second visualization uses a bar graph (sns.countplot) to show the comparison of the number of passengers who survived (Survived) based on gender.

VISUALIZATION

- **sns.histplot(...)** : Creates a histogram using Seaborn to see the distribution of values in the Age column.
- **df['Age']** : The column to be visualized, namely the age of the passengers.
- **bins=30** : Divides the age range into 30 age groups (intervals), so that the distribution is more detailed.
- **plt.title(...)** : Adds a title to the graph to make it easier to understand.
- **plt.show()** : Displays the graph.

```
[ ]  sns.histplot(df['Age'], bins=30)
     plt.title('Distribusi Usia Penumpang')
     plt.show()
```

```
sns.countplot(x='Survived', hue='Sex', data=df)
plt.title('Survival berdasarkan Gender')
plt.show()
```

The code displays a bar graph (countplot) to compare the number of passengers who survived (Survived) based on gender (Sex). With the hue='Sex' parameter, the graph shows the difference between men and women in terms of survival rate.

THANK YOU