

```
In [29]: import pandas as pd
import seaborn as sns

# Read the Data with Pandas
df = pd.read_csv("train.csv")
df
```

Out[29]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8
...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7

891 rows × 12 columns

```
In [31]: # 1. Create df1: Identification and Outcome
# Selecting specifically from the columns you listed
df1 = df[['PassengerId', 'Name', 'Age', 'Survived']].copy()

# 2. Create df2: Socio-Economic and Travel Data
df2 = df[['PassengerId', 'Pclass', 'Sex', 'Fare']].copy()

print("DataFrames successfully created!")
display(df1.head())
display(df2.head())
```

DataFrames successfully created!

	PassengerId	Name	Age	Survived
0	1	Braund, Mr. Owen Harris	22.0	0
1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1
2	3	Heikkinen, Miss. Laina	26.0	1
3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1
4	5	Allen, Mr. William Henry	35.0	0

	PassengerId	Pclass	Sex	Fare
0	1	3	male	7.2500
1	2	1	female	71.2833
2	3	3	female	7.9250
3	4	1	female	53.1000
4	5	3	male	8.0500

```
In [41]: # Inner Join on 'PassengerId'
merged_inner = pd.merge(df1, df2, on='PassengerId', how='inner')
print("\nInner Join on 'PassengerId':\n")
merged_inner.head()
```

Inner Join on 'PassengerId':

Out[41]:	PassengerId	Name	Age	Survived	Pclass	Sex	Fare
0	1	Braund, Mr. Owen Harris	22.0	0	3	male	7.2500
1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	1	female	71.2833
2	3	Heikkinen, Miss. Laina	26.0	1	3	female	7.9250
3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	1	female	53.1000
4	5	Allen, Mr. William Henry	35.0	0	3	male	8.0500

```
In [43]: # Left Join on 'PassengerId'
merged_left = pd.merge(df1, df2, on='PassengerId', how='left')
print("\nLeft Join on 'PassengerId':\n")
merged_left.head()
```

Left Join on 'PassengerId':

Out[43]:	PassengerId	Name	Age	Survived	Pclass	Sex	Fare
0	1	Braund, Mr. Owen Harris	22.0	0	3	male	7.2500
1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	1	female	71.2833
2	3	Heikkinen, Miss. Laina	26.0	1	3	female	7.9250
3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	1	female	53.1000
4	5	Allen, Mr. William Henry	35.0	0	3	male	8.0500

```
In [45]: # Right Join on 'passengPassengerId_id'
merged_right = pd.merge(df1, df2, on='PassengerId', how='right')
```

```
print("\nRight Join on 'PassengerId':\n")
merged_right.head()
```

Right Join on 'PassengerId':

Out[45]:

	PassengerId	Name	Age	Survived	Pclass	Sex	Fare
0	1	Braund, Mr. Owen Harris	22.0	0	3	male	7.2500
1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	1	female	71.2833
2	3	Heikkinen, Miss. Laina	26.0	1	3	female	7.9250
3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	1	female	53.1000
4	5	Allen, Mr. William Henry	35.0	0	3	male	8.0500

In [47]:

```
# Outer Join on 'passenger_id'
merged_outer = pd.merge(df1, df2, on='PassengerId', how='outer')
print("\nOuter Join on 'PassengerId':\n")
merged_outer.head()
```

Outer Join on 'PassengerId':

Out[47]:

	PassengerId	Name	Age	Survived	Pclass	Sex	Fare
0	1	Braund, Mr. Owen Harris	22.0	0	3	male	7.2500
1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	1	female	71.2833
2	3	Heikkinen, Miss. Laina	26.0	1	3	female	7.9250
3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	1	female	53.1000
4	5	Allen, Mr. William Henry	35.0	0	3	male	8.0500

In [49]:

```
# Concatenate without keys (default behavior)
concatenated = pd.concat([df1, df2])
print("\nConcatenation along rows:\n")
concatenated.head()
```

Concatenation along rows:

Out[49]:

	PassengerId	Name	Age	Survived	Pclass	Sex	Fare
0	1	Braund, Mr. Owen Harris	22.0	0.0	NaN	NaN	NaN
1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1.0	NaN	NaN	NaN
2	3	Heikkinen, Miss. Laina	26.0	1.0	NaN	NaN	NaN
3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1.0	NaN	NaN	NaN
4	5	Allen, Mr. William Henry	35.0	0.0	NaN	NaN	NaN

In [51]:

```
# Concatenate with specific keys
concatenated_with_keys = pd.concat([df1, df2], keys=['df1', 'df2'])
print("\nConcatenation with keys:\n")
concatenated_with_keys.head()
```

Concatenation with keys:

Out[51]:

	PassengerId	Name	Age	Survived	Pclass	Sex	Fare	
df1	0	1	Braund, Mr. Owen Harris	22.0	0.0	NaN	NaN	NaN
	1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1.0	NaN	NaN	NaN
	2	3	Heikkinen, Miss. Laina	26.0	1.0	NaN	NaN	NaN
	3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1.0	NaN	NaN	NaN
	4	5	Allen, Mr. William Henry	35.0	0.0	NaN	NaN	NaN

In [53]:

```
# Concatenate along columns (axis=1)
concatenated_axis1 = pd.concat([df1, df2], axis=1)
print("\nConcatenation along columns (axis=1):\n")
concatenated_axis1.head()
```

Concatenation along columns (axis=1):

Out[53]:

	PassengerId	Name	Age	Survived	PassengerId	Pclass	Sex	Fare
0	1	Braund, Mr. Owen Harris	22.0	0	1	3	male	7.2500
1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	2	1	female	71.2833
2	3	Heikkinen, Miss. Laina	26.0	1	3	3	female	7.9250
3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	4	1	female	53.1000
4	5	Allen, Mr. William Henry	35.0	0	5	3	male	8.0500

In [73]:

```
# Calculate mean, median, and mode for numerical columns
print("\n----- Mean ----- \n", titanic[['age', 'fare']].mean())
print("\n----- Median ----- \n", titanic[['age', 'fare']].median())
print("\n----- Mode ----- \n", titanic[['age', 'fare']].mode())
```

----- Mean -----
age 29.699118
fare 32.204208
dtype: float64

----- Median -----
age 28.0000
fare 14.4542
dtype: float64

----- Mode -----
age fare
0 24.0 8.05

In [67]:

```
# Calculate mid-range (average of max and min for each column)
mid_range = titanic[['age', 'fare']].apply(lambda x: (x.max() + x.min()) / 2)
print("\n----- Mid-Range ----- \n", mid_range)
```

----- Mid-Range -----
age 40.2100
fare 256.1646
dtype: float64