```
%matplotlib inline
```

```
import pandas as pd

url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"
df = pd.read_csv(url)

df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.info()
df.describe(include='all')
```

```
<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
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Pa
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000
unique	NaN	NaN	NaN	891	2	NaN	NaN	
top	NaN	NaN	NaN	Dooley, Mr. Patrick	male	NaN	NaN	
freq	NaN	NaN	NaN	1	577	NaN	NaN	
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000
max	891.000000	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.000

Most columns have ≤ 891 non-null values; Age and Cabin have missing data.

```
# Missing values per column
df.isna().sum().sort_values(ascending=False)
```

	0
Cabin	687
Age	177
Embarked	2
PassengerId	0
Name	0
Pclass	0
Survived	0
Sex	0
Parch	0
SibSp	0
Fare	0
Ticket	0

dtype: int64

note which columns need attention (Age, Cabin, Embarked).

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

# Age distribution
sns.histplot(df['age'].dropna(), kde=True)
plt.title('Age Distribution'); plt.show()

# Survival counts
sns.countplot(x='survived', data=df)
plt.title('Survival Count (0 = Died, 1 = Survived)'); plt.show()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.11/dist-packages/pandas/core/indexes/base.py in
get_loc(self, key)
   3804         try:
-> 3805             return self._engine.get_loc(casted_key)
   3806         except KeyError as err:

index.pyx in pandas._libs.index.IndexEngine.get_loc()

index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'age'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
                                  2 frames
/usr/local/lib/python3.11/dist-packages/pandas/core/indexes/base.py in
get_loc(self, key)
   3810             ):
   3811                 raise InvalidIndexError(key)
-> 3812             raise KeyError(key) from err
   3813         except TypeError:
   3814             # If we have a listlike key, _check_indexing_error will
raise

KeyError: 'age'
```

Next steps: Explain error

```
print(df.columns)
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

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

# Change column names if needed based on Step 1
sns.histplot(df['age'].dropna(), kde=True)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()

sns.countplot(x='survived', data=df)
plt.title('Survival Count (0 = Died, 1 = Survived)')
plt.xlabel('Survived')
plt.ylabel('Number of Passengers')
plt.show()
```

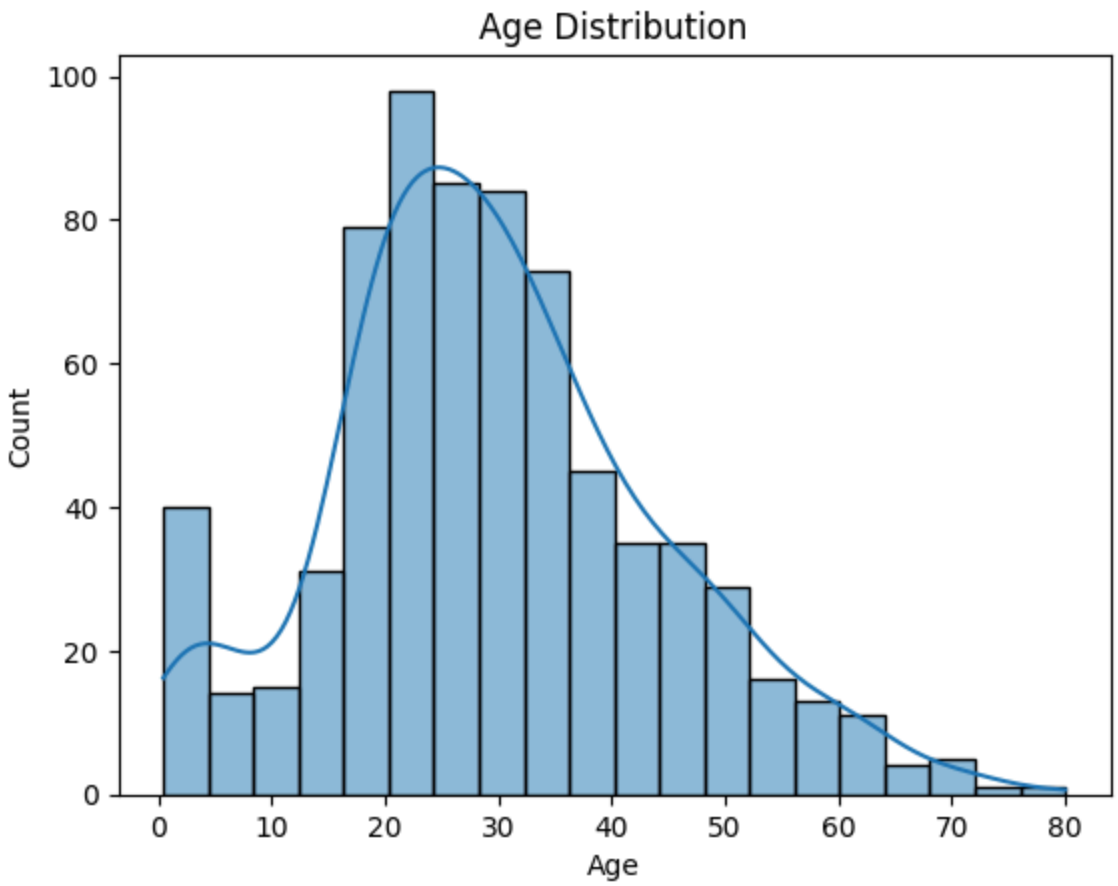
```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.11/dist-packages/pandas/core/indexes/base.py in
get_loc(self, key)
   3804         try:
-> 3805             return self._engine.get_loc(casted_key)
   3806         except KeyError as err:

index.pyx in pandas._libs.index.IndexEngine.get_loc()

index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'age'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
                                  2 frames
/usr/local/lib/python3.11/dist-packages/pandas/core/indexes/base.py in
get_loc(self, key)
   3810             ):
   3811                 raise InvalidIndexError(key)
-> 3812             raise KeyError(key) from err
   3813         except TypeError:
   3814             # If we have a listlike key, _check_indexing_error will
raise

KeyError: 'age'
```

Next steps: Explain error

```
print(df['Age'].dropna().head())        # Check some ages
print(df['Survived'].value_counts())    # Count survived vs not
```

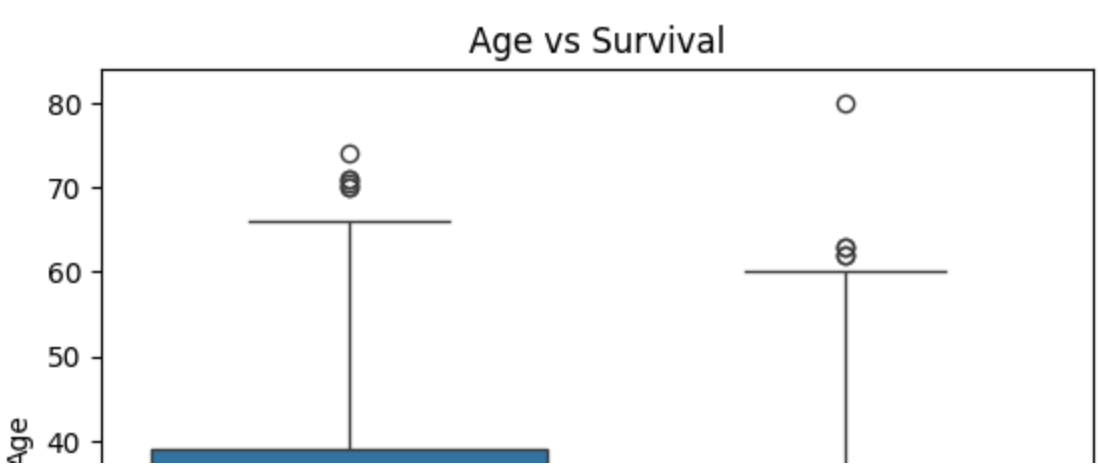
```
0    22.0
1    38.0
2    26.0
3    35.0
4    35.0
Name: Age, dtype: float64
Survived
0    549
1    342
Name: count, dtype: int64
```

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

# Plot 1: Age distribution
sns.histplot(df['Age'].dropna(), kde=True)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()

# Plot 2: Survival count
sns.countplot(x='Survived', data=df)
plt.title('Survival Count (0 = Died, 1 = Survived)')
plt.xlabel('Survived')
plt.ylabel('Number of Passengers')
plt.show()

print("✅ Plots finished")
```

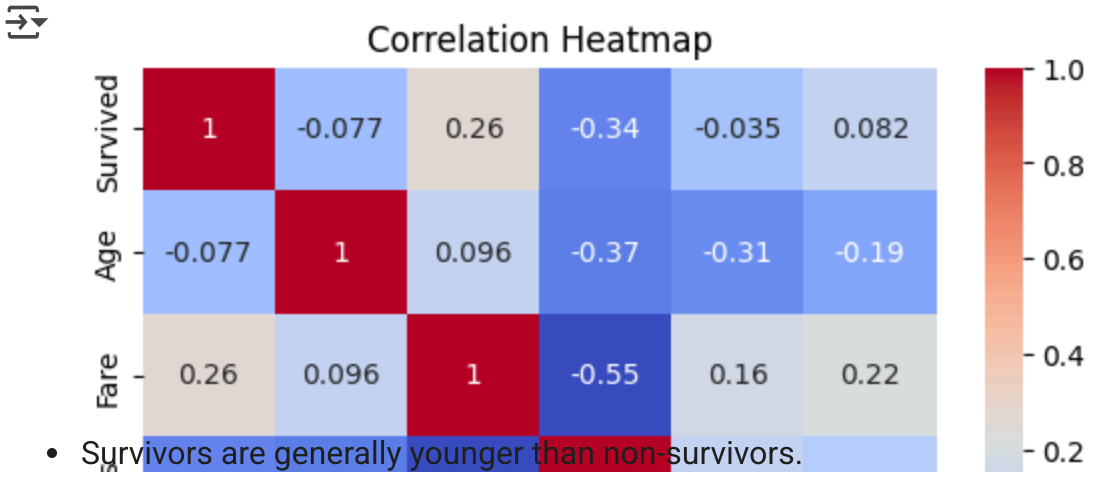


- Most passengers are between 20 and 40 years old.
- Around 550 passengers did not survive; about 340 survived.

```
# Boxplot: Age vs Survived
sns.boxplot(x='Survived', y='Age', data=df)
plt.title('Age vs Survival')
plt.xlabel('Survived')
plt.ylabel('Age')
plt.show()
```



```
# Select only numeric columns for correlation
numeric_cols = ['Survived', 'Age', 'Fare', 'Pclass', 'SibSp', 'Parch']
corr = df[numeric_cols].corr()

# Plot heatmap
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



- Survivors are generally younger than non-survivors.
- Fare is positively correlated with survival.
- Pclass has a negative correlation with survival (1st class survived more).

```
# Pairplot (may take ~20-30 s to render on mobile)
pair_cols = ['Survived', 'Age', 'Fare', 'Pclass']
sns.pairplot(df[pair_cols], hue='Survived')
plt.suptitle('Pairplot of Key Variables', y=1.02)
plt.show()
```