In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm
```

In [4]:
```
url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/tit
df = pd.read_csv(url)
```

In [5]:
```
df.head()
```

Out[5]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	I
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

In [6]:
```
# Fill missing 'Age' with the median value
df['Age'].fillna(df['Age'].median(), inplace=True)

# Fill missing 'Embarked' with the most common value
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)

# Drop the 'Cabin' column as it has too many missing values
df.drop('Cabin', axis=1, inplace=True)

# Verify no missing values remain
df.isnull().sum()
```

```
/tmp/ipykernel_6719/3264089490.py:2: FutureWarning: A value is trying to be s
et on a copy of a DataFrame or Series through chained assignment using an inp
lace method.
The behavior will change in pandas 3.0. This inplace method will never work b
ecause the intermediate object on which we are setting values always behaves
as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'd
f.method({col: value}, inplace=True)' or df[col] = df[col].method(value) inst
ead, to perform the operation inplace on the original object.


  df['Age'].fillna(df['Age'].median(), inplace=True)
/tmp/ipykernel_6719/3264089490.py:5: FutureWarning: A value is trying to be s
et on a copy of a DataFrame or Series through chained assignment using an inp
lace method.
The behavior will change in pandas 3.0. This inplace method will never work b
ecause the intermediate object on which we are setting values always behaves
as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'd
f.method({col: value}, inplace=True)' or df[col] = df[col].method(value) inst
ead, to perform the operation inplace on the original object.


  df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
```

Out[6]:

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

In [7]:

```
# Convert 'Sex' to numerical values (male: 0, female: 1)
df['Sex'] = df['Sex'].map({'male': 0, 'female': 1})

# One-hot encode 'Embarked'
df = pd.get_dummies(df, columns=['Embarked'], drop_first=True)

# Display the updated dataset
df.head()
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Far
0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.250
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.283
2	3	1	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/ O2. 3101282	7.925
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.0	1	0	113803	53.100
4	5	0	3	Allen, Mr. William Henry	0	35.0	0	0	373450	8.050

In [8]:
```
# Features (independent variables)
X = df[['Fare', 'Pclass']]

# Target (dependent variable)
y = df['Age']
```

In [9]:
```
# Split the data (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran
```

In [10]:
```
# Create and train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Display the coefficients
print(f"Intercept (β0): {model.intercept_}")
print(f"Coefficients (β1, β2): {model.coef_}")
```

```
Intercept (β0): 44.937186049769814
Coefficients (β1, β2): [-0.03201404 -6.30449967]
```

In [11]:
```
# Predict on the test set
y_pred = model.predict(X_test)

# Display the first few predictions
print(y_pred[:5])
```

```
[25.53560736 31.99203927 25.76997576 31.27172331 25.66379479]
```