```python
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

```python
# Replace 'path_to_your_dataset.csv' with the path to your actual dataset file
dataset = pd.read_csv('/content/salary_data.csv')
```

```python
dataset
```

	YearsExperience	Age	Salary
0	1.1	21.0	39343
1	1.3	21.5	46205
2	1.5	21.7	37731
3	2.0	22.0	43525
4	2.2	22.2	39891
5	2.9	23.0	56642
6	3.0	23.0	60150
7	3.2	23.3	54445
8	3.2	23.3	64445
9	3.7	23.6	57189
10	3.9	23.9	63218
11	4.0	24.0	55794
12	4.0	24.0	56957
13	4.1	24.0	57081
14	4.5	25.0	61111
15	4.9	25.0	67938
16	5.1	26.0	66029
17	5.3	27.0	83088
18	5.9	28.0	81363
19	6.0	29.0	93940
20	6.8	30.0	91738
21	7.1	30.0	98273
22	7.9	31.0	101302
23	8.2	32.0	113812
24	8.7	33.0	109431
25	9.0	34.0	105582
26	9.5	35.0	116969
27	9.6	36.0	112635
28	10.3	37.0	122391
29	10.5	38.0	121872

Next steps: Generate code with `dataset`   ● View recommended plots   New interactive sheet

```
# Replace 'YearsExperience' and 'Salary' with the actual column names from your dataset
X = dataset[['YearsExperience']]  # Independent variable(s)
y = dataset['Salary']             # Dependent variable
```

```
# Step 4: Splitting the dataset into the Training set and Test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
# Step 5: Fitting Simple Linear Regression to the Training set
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
▾ LinearRegression
  LinearRegression()
```

```
# Step 6: Predicting the Test set results
y_pred = regressor.predict(X_test)
```

```
# Step 7: Visualizing the Training set results
plt.scatter(X_train, y_train, color='red')  # Real data points
plt.plot(X_train, regressor.predict(X_train), color='blue')  # Regression line
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```