

✓ Simple Linear Regression

```
from google.colab import drive
drive.mount('/content/drive')
```

⇨ Mounted at /content/drive

✓ Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

✓ Importing the dataset

```
dataset = pd.read_csv('/content/drive/MyDrive/AI Tools Lab/Salary_Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
print(X.shape)
print(y.shape)
print(X)
print(y)
```

⇨ (30, 1)
(30,)
[[1.1]
[1.3]
[1.5]
[2.]
[2.2]
[2.9]
[3.]
[3.2]
[3.2]
[3.7]
[3.9]
[4.]
[4.]
[4.1]
[4.5]
[4.9]
[5.1]
[5.3]
[5.9]
[6.]
[6.8]
[7.1]
[7.9]

```
[ 8.2]
[ 8.7]
[ 9. ]
[ 9.5]
[ 9.6]
[10.3]
[10.5]]
[ 39343  46205  37731  43525  39891  56642  60150  54445  64445  57189
 63218  55794  56957  57081  61111  67938  66029  83088  81363  93940
 91738  98273 101302 113812 109431 105582 116969 112635 122391 121872]
```

✓ Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state =
```

✓ Training the Simple Linear Regression model on the Training set

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```



```
LinearRegression
LinearRegression()
```

✓ Predicting the Test set results

```
y_pred = regressor.predict(X_test)
```

✓ Visualising the Training set results

```
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

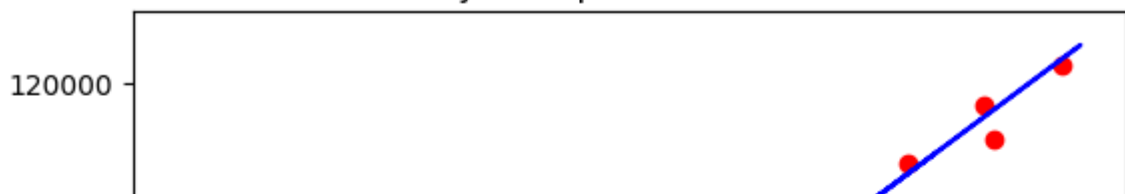


✓ Visualising the Test set results

```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```



Salary vs Experience (Test set)



```
# to check whether the model works satisfactorily,  
# obtain the coefficient of determination,  $R^2$ , with .score()  
r_sq = regressor.score(X_test, y_test)  
print(f"coefficient of determination: {r_sq}")
```



coefficient of determination: 0.9749154407708353

```
# r2_score and regressor.score, both yield same output.  
from sklearn.metrics import r2_score  
r2=r2_score(y_test, y_pred)  
print(r2)
```



0.9749154407708353

Start coding or [generate](#) with AI.