Simple Linear Regression

Importing the libraries

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importing the dataset

In [2]:
```
dataset = pd.read_csv('Salary_Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

Splitting the dataset into the Training set and Test set

In [3]:
```
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 = 0)
```

Training the Simple Linear Regression model on the Training set

In [4]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[4]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

Predicting the Test set results

In [5]:
```
y_pred = regressor.predict(X_test)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-5-29d4074f386a> in <module>()
----> 1 y_pred = regressor.predict(X_test)

C:\Users\jithe\Anaconda2\lib\site-packages\sklearn\linear_model\base.pyc in predict(self, X)
    211             Returns predicted values.
    212         """
--> 213         return self._decision_function(X)
    214
    215     _preprocess_data = staticmethod(_preprocess_data)

C:\Users\jithe\Anaconda2\lib\site-packages\sklearn\linear_model\base.pyc in _decision_function(self, X)
    194         check_is_fitted(self, "coef_")
    195
--> 196         X = check_array(X, accept_sparse=['csr', 'csc', 'coo'])
    197         return safe_sparse_dot(X, self.coef_.T,
    198                                dense_output=True) + self.intercept_

C:\Users\jithe\Anaconda2\lib\site-packages\sklearn\utils\validation.pyc in check_array(array, accept_sparse, accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allow_nd, ensure_min_samples, ensure_min_features, warn_on_dtype, estimator)
    580                                 " minimum of %d is required%s."
    581                                 % (n_samples, shape_repr, ensure_min_samples,
--> 582                                    context))
    583
    584     if ensure_min_features > 0 and array.ndim == 2:

ValueError: Found array with 0 sample(s) (shape=(0, 1)) while a minimum of 1 is required.
```

Visualising the Training set results

In [6]:
```
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

In [7]:
```
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()
```