Simple Linear Regression

import library

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

import data set

```
df = pd.read_csv('FuelConsumption.csv')

df.head()
```

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	TRANSMISSION
0	2014	ACURA	ILX	COMPACT	2.0	4	AS5
1	2014	ACURA	ILX	COMPACT	2.4	4	M6
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6

```
df = df[['ENGINESIZE', 'CO2EMISSIONS']]

df.head()
```

	ENGINESIZE	CO2EMISSIONS
0	2.0	196
1	2.4	221
2	1.5	136
3	3.5	255
4	3.5	244

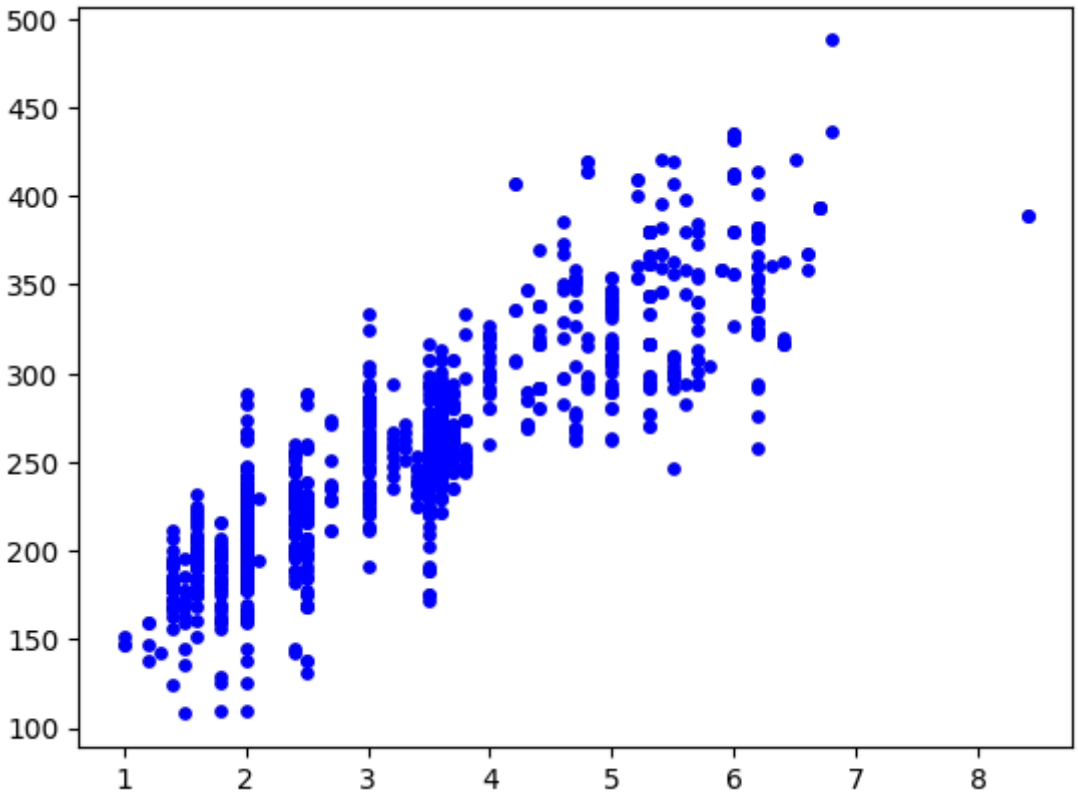
```
# df.size
# df.shape
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1067 entries, 0 to 1066
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   ENGINESIZE    1067 non-null   float64
 1   CO2EMISSIONS  1067 non-null   int64  
dtypes: float64(1), int64(1)
memory usage: 16.8 KB
```

```
df.describe()
```

	ENGINESIZE	CO2EMISSIONS
count	1067.000000	1067.000000
mean	3.346298	256.228679
std	1.415895	63.372304
min	1.000000	108.000000
25%	2.000000	207.000000
50%	3.400000	251.000000
75%	4.300000	294.000000
max	8.400000	488.000000

```
plt.scatter(df['ENGINESIZE'], df['CO2EMISSIONS'],s=15, color='b
plt.show()
```



cleaning the data

```
# all data are not null
```

encoding the data

```
# no need for encoding
```

define x , y

```
x = np.array(df[['ENGINESIZE']])
y = np.array(df[['CO2EMISSIONS']])
```

```
x[:5]
```

```
array([[2. ],
       [2.4],
       [1.5],
       [3.5],
       [3.5]])
```

```
# x = df['ENGINESIZE'].values
# y = df['CO2EMISSIONS'].values
```

```
# x
```

```
array([2. , 2.4, 1.5, ..., 3. , 3.2, 3.2])
```

```
# y
```

```
array([196, 221, 136, ..., 271, 260, 294], dtype=int64)
```

spliting the data

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_
```

```
print(x_train.shape)
print(x_test.shape)
```

```
(853, 1)
(214, 1)
```

```
# msk = np.random.rand(len(df)) < 0.8
# train = df[msk]
# test = df[~msk]
```

```
# x_train = np.array(train[['ENGINESIZE']])
# x_test = np.array(test[['ENGINESIZE']])
# y_train = np.array(train[['CO2EMISSIONS']])
# y_test = np.array(test[['CO2EMISSIONS']])
```

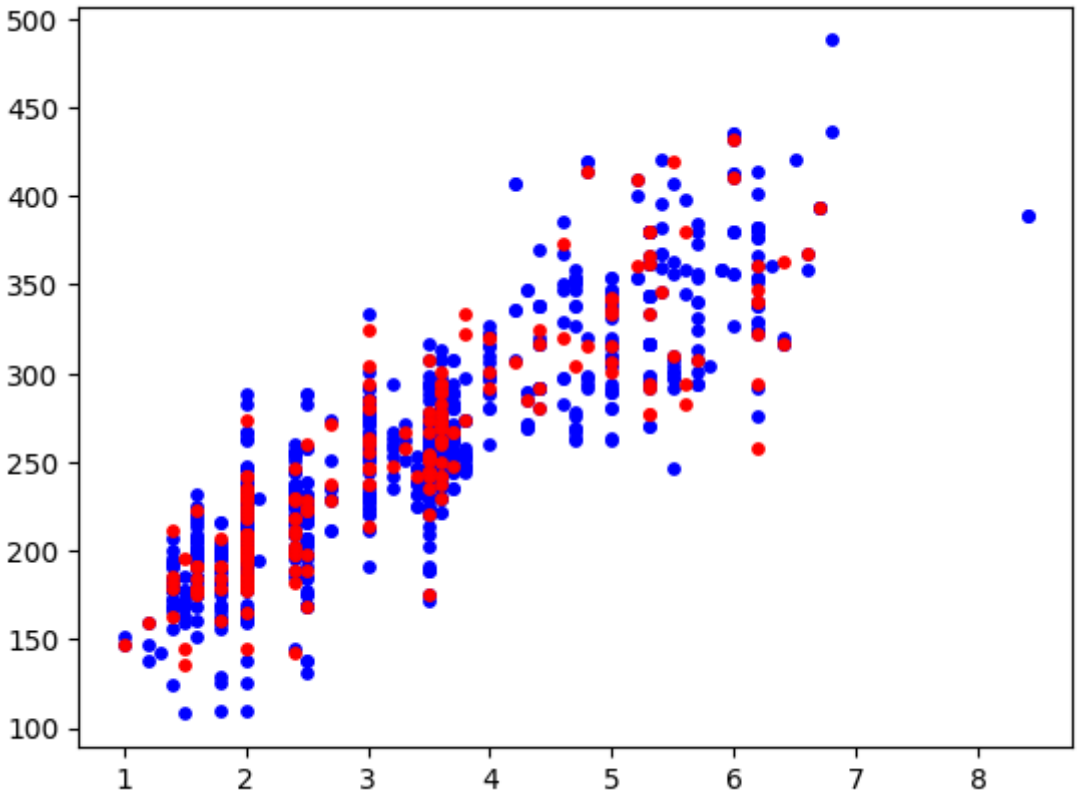
```
# x_train.size
```

```
854
```

```
# x_test.size
```

```
213
```

```
plt.scatter(x_train, y_train, s=15, c='b')
plt.scatter(x_test, y_test, s=15, c='r')
plt.show()
```



fit train data

```
from sklearn.linear_model import LinearRegression
slr = LinearRegression()
```

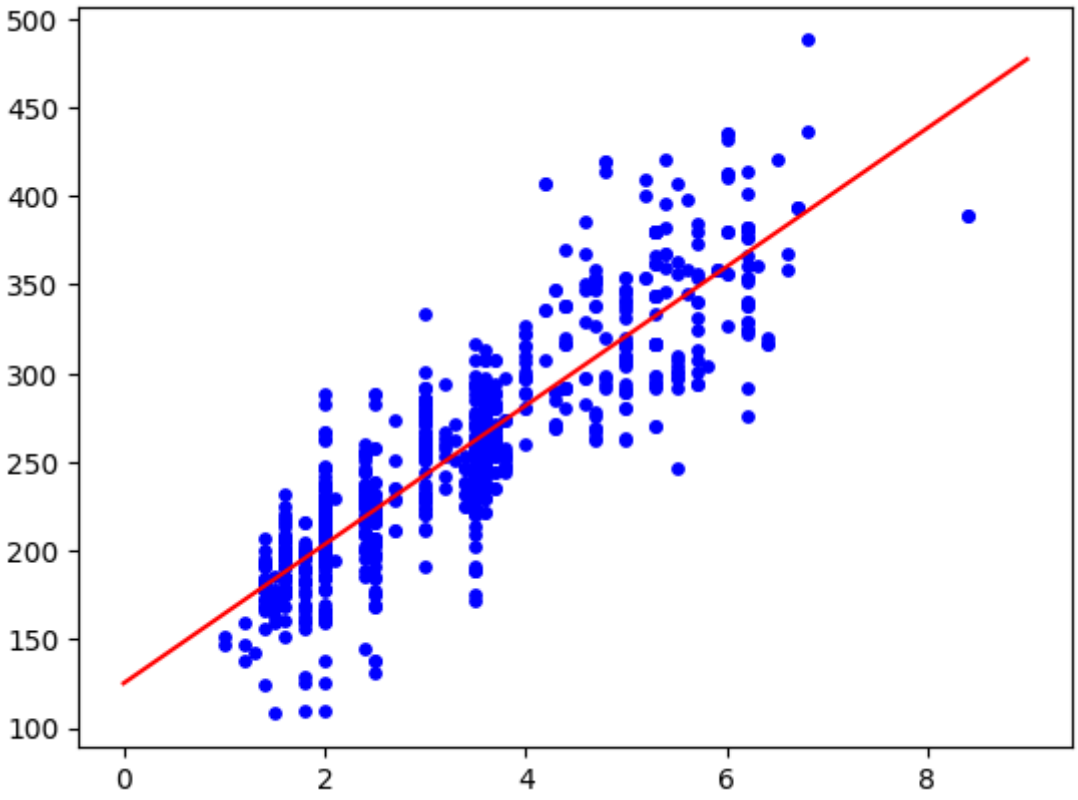
```
slr.fit(x_train, y_train)
```

```
LinearRegression()
```

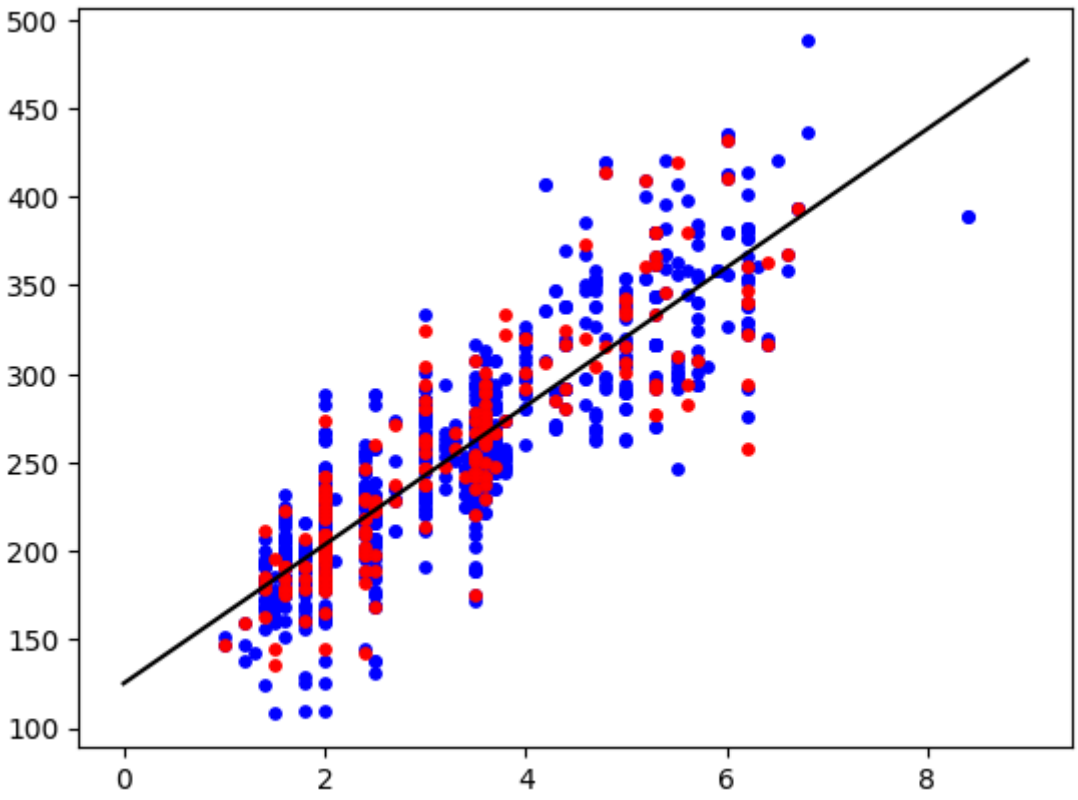
```
print(slr.intercept_)
print(slr.coef_)
```

```
[125.16103129]
[[39.15878276]]
```

```
xx = np.arange(0,9,0.01)
plt.scatter(x_train, y_train, s=15, c='b')
plt.plot(xx, slr.intercept_[0] + slr.coef_[0] * xx, c='r')
plt.show()
```




```
xx = np.arange(0,9,0.01)
plt.scatter(x_train, y_train, s=15, c='b')
plt.plot(xx, slr.intercept_[0] + slr.coef_[0] * xx, c='black')
plt.scatter(x_test, y_test, c='r', s=15)
plt.show()
```



predict test data

```
yhat_test = slr.predict(x_test)
```

evaluate the model

```
from sklearn.metrics import r2_score
print("r2-score: %0.2f" % r2_score(y_test, yhat_test))
```

```
r2-score: 0.76
```

```
# print("MAE: %0.2f" % np.mean(np.absolute(y_test - yhat_test))
# print("MSE: %0.2f" % np.mean((y_test - yhat_test) ** 2))
```

```
MAE: 23.67
MSE: 968.42
```

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test, yhat_test)
mean_squared_error(y_test, yhat_test)
```

```
926.2716910067601
```

predict new data

```
slr.predict([[0]])
```

```
array([[125.16103129]])
```