Simple Linear Regression

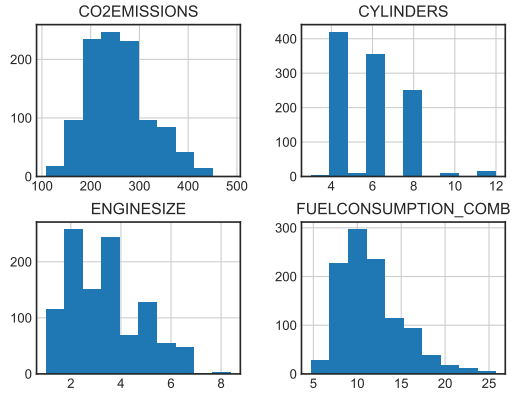
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rc, font_manager
from sklearn import linear_model

ticks_font = font_manager.FontProperties(family='Times New Roman', style='normal',
    size=12, weight='normal', stretch='normal')
plt.style.use('seaborn-white')
ax=plt.gca()
## Loading Data ##
df=pd.read_csv('D:\Python\edx\Machine Learning\FuelConsumptionCo2.csv')
with open('LinearReg.txt','a') as f:
    print(df.head(),file=f)
    print(df.describe(),file=f)

f_col=['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB','CO2EMISSIONS']
X=df[f_col]
with open('LinearReg.txt','a') as f:
    print(X.head(),file=f)

## Histogram ##

viz=X[['CYLINDERS','ENGINESIZE','CO2EMISSIONS','FUELCONSUMPTION_COMB']]
viz.hist()


## Scatter Plot vs CO2 emissions ##
plt.figure()
plt.scatter(X.FUELCONSUMPTION_COMB, X.CO2EMISSIONS,color='blue')
plt.title('Scatter Plot - Fuel Consumption vs Emissions',fontname='Times New Roman',
          fontsize=12)
plt.ylabel('Emissions',fontname='Times New Roman',fontsize=12)
plt.xlabel('Fuel Consumption',fontname='Times New Roman',fontsize=12)

plt.figure()
plt.scatter(X.ENGINESIZE, X.CO2EMISSIONS,color='blue')
plt.ylabel('Emissions',fontname='Times New Roman',fontsize=12)
plt.xlabel('Engine Size',fontname='Times New Roman',fontsize=12)

plt.figure()
```

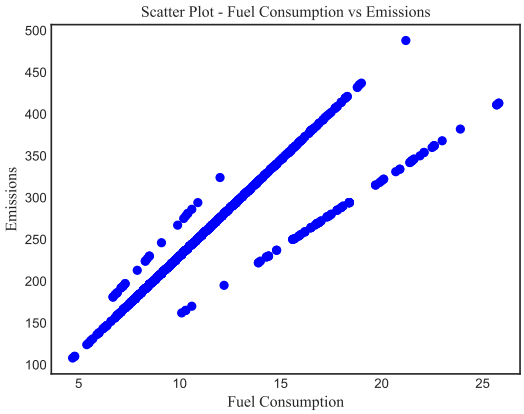
```
plt.scatter(X.CYLINDERS, X.CO2EMISSIONS,color='blue')
plt.ylabel('Emissions',fontname='Times New Roman',fontsize=12)
plt.xlabel('Number of Cylinders',fontname='Times New Roman',fontsize=12)

## Train/Test Split Data ##
mask=np.random.rand(len(df))< 0.8
train=X[mask]
test=X[~mask]

##Train Distr. ##
plt.figure()
plt.scatter(train.ENGINESIZE, train.CO2EMISSIONS,color='blue')
plt.ylabel('Emissions',fontname='Times New Roman',fontsize=12)
plt.xlabel('Engine Size',fontname='Times New Roman',fontsize=12)

Lreg=linear_model.LinearRegression()
train_x=np.asanyarray(train[['ENGINESIZE']])
train_y=np.asanyarray(train[['CO2EMISSIONS']])
Lreg.fit(train_x,train_y)
with open('LinearReg.txt','a') as f:
    print('Coeff: ', Lreg.coef_,file=f)
    print('Intercept: ',Lreg.intercept_,file=f)

### With fit line ##
plt.figure()
plt.scatter(train.ENGINESIZE, train.CO2EMISSIONS,  color='blue')
plt.plot(train_x, Lreg.coef_[0][0]*train_x + Lreg.intercept_[0], '-r')
plt.ylabel('Emissions',fontname='Times New Roman',fontsize=12)
plt.xlabel('Engine Size',fontname='Times New Roman',fontsize=12)

### Evaluation ###

from sklearn.metrics import r2_score

test_x = np.asanyarray(test[['ENGINESIZE']])
test_y = np.asanyarray(test[['CO2EMISSIONS']])
test_y_ = Lreg.predict(test_x)
with open('LinearReg.txt','a') as f:
    print("Mean absolute error: %.2f" % np.mean(np.absolute(test_y_ - test_y)),file=f)
    print("Residual sum of squares (MSE): %.2f" % np.mean((test_y_ - test_y) ** 2),file=
    print("R2-score: %.2f" % r2_score(test_y_ , test_y),file=f )

##Display plot##
for label in ax.get_xticklabels():
    label.set_fontproperties(ticks_font)
```

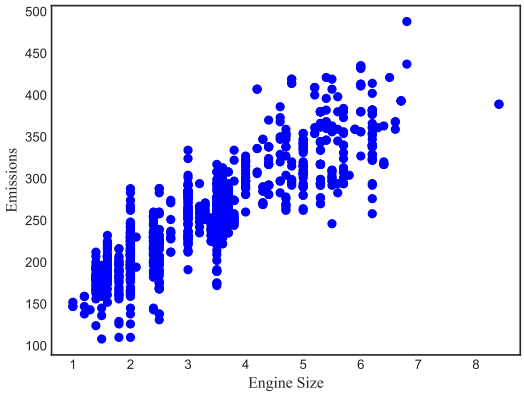
```
for label in ax.get_yticklabels():
    label.set_fontproperties(ticks_font)
plt.show()
```

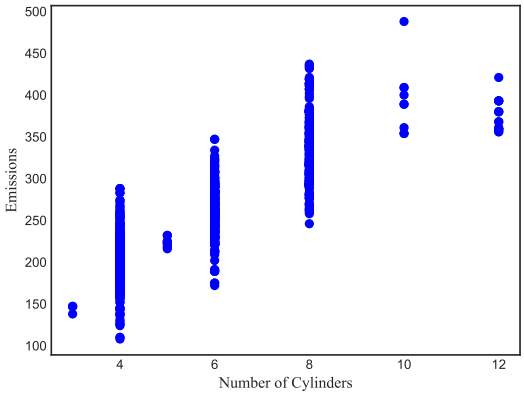
Solution:

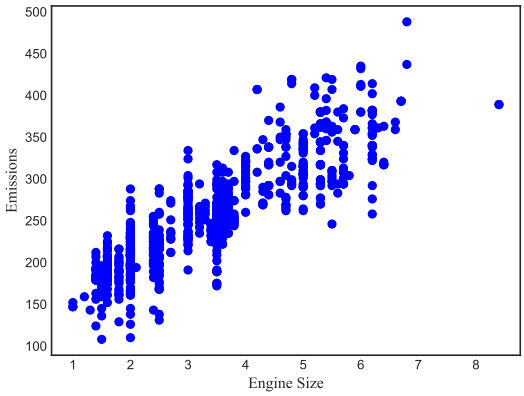
```
   MODELYEAR   MAKE  ... FUELCONSUMPTION_COMB_MPG CO2EMISSIONS
0       2014  ACURA  ...                      33          196
1       2014  ACURA  ...                      29          221
2       2014  ACURA  ...                      48          136
3       2014  ACURA  ...                      25          255
4       2014  ACURA  ...                      27          244

[5 rows x 13 columns]
       MODELYEAR   ENGINESIZE  ...  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
count     1067.0  1067.000000  ...               1067.000000   1067.000000
mean      2014.0     3.346298  ...                 26.441425    256.228679
std          0.0     1.415895  ...                  7.468702     63.372304
min       2014.0     1.000000  ...                 11.000000    108.000000
25%       2014.0     2.000000  ...                 21.000000    207.000000
50%       2014.0     3.400000  ...                 26.000000    251.000000
75%       2014.0     4.300000  ...                 31.000000    294.000000
max       2014.0     8.400000  ...                 60.000000    488.000000

[8 rows x 8 columns]
   ENGINESIZE  CYLINDERS  FUELCONSUMPTION_COMB  CO2EMISSIONS
0         2.0          4                   8.5           196
1         2.4          4                   9.6           221
2         1.5          4                   5.9           136
3         3.5          6                  11.1           255
4         3.5          6                  10.6           244
Coeff:  [[38.59248093]]
Intercept:  [126.77496723]
Mean absolute error: 22.38
Residual sum of squares (MSE): 863.51
R2-score: 0.71
```

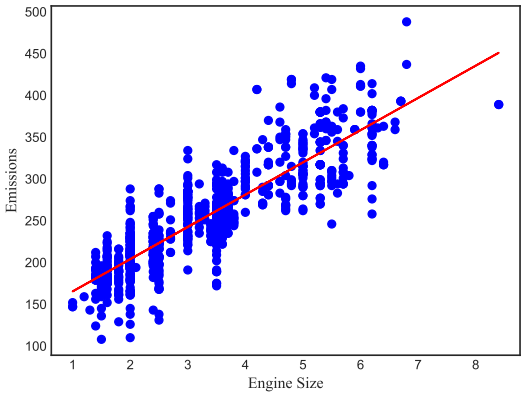

CO2EMISSIONS
CYLINDERS
ENGINESIZE
FUELCONSUMPTION_COMB


Scatter Plot - Fuel Consumption vs Emissions
Emissions
Fuel Consumption


Emissions
Engine Size


Emissions
Number of Cylinders


Emissions
Engine Size


Emissions
Engine Size