[10]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import pylab as pl
%matplotlib inline
```

[11]:
```
df= pd.read_csv("FuelConsumption.csv")
```

[12]:
```
df.head()
```

[12]:

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

	TRANSMISSION	FUELTYPE	FUELCONSUMPTION_CITY	FUELCONSUMPTION_HWY	\
0	AS5	Z	9.9	6.7	
1	M6	Z	11.2	7.7	
2	AV7	Z	6.0	5.8	
3	AS6	Z	12.7	9.1	
4	AS6	Z	12.1	8.7	

	FUELCONSUMPTION_COMB	FUELCONSUMPTION_COMB_MPG	CO2EMISSIONS
0	8.5	33	196
1	9.6	29	221
2	5.9	48	136
3	11.1	25	255
4	10.6	27	244

[13]:
```
df.describe()
```

[13]:

	MODELYEAR	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_CITY	\
count	1067.0	1067.000000	1067.000000	1067.000000	
mean	2014.0	3.346298	5.794752	13.296532	
std	0.0	1.415895	1.797447	4.101253	

```
min       2014.0     1.000000     3.000000              4.600000
25%       2014.0     2.000000     4.000000             10.250000
50%       2014.0     3.400000     6.000000             12.600000
75%       2014.0     4.300000     8.000000             15.550000
max       2014.0     8.400000    12.000000             30.200000

       FUELCONSUMPTION_HWY  FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  \
count          1067.000000           1067.000000               1067.000000
mean              9.474602             11.580881                 26.441425
std               2.794510              3.485595                  7.468702
min               4.900000              4.700000                 11.000000
25%               7.500000              9.000000                 21.000000
50%               8.800000             10.900000                 26.000000
75%              10.850000             13.350000                 31.000000
max              20.500000             25.800000                 60.000000

       CO2EMISSIONS
count   1067.000000
mean     256.228679
std       63.372304
min      108.000000
25%      207.000000
50%      251.000000
75%      294.000000
max      488.000000
```

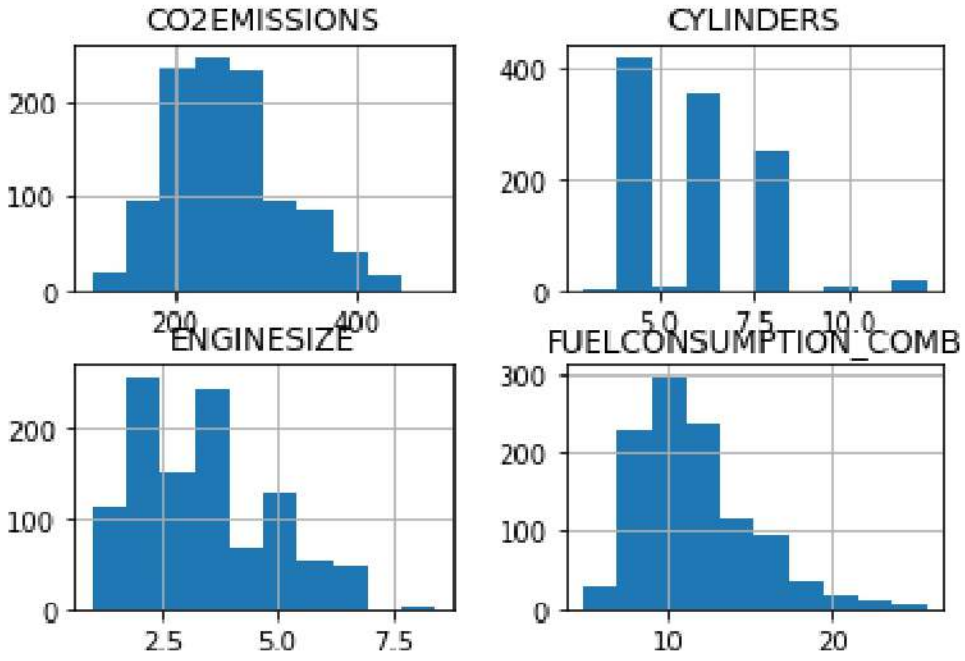
[14]:
```
cdf=df[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB','CO2EMISSIONS']]
```

[15]:
```
cdf.head(9)
```

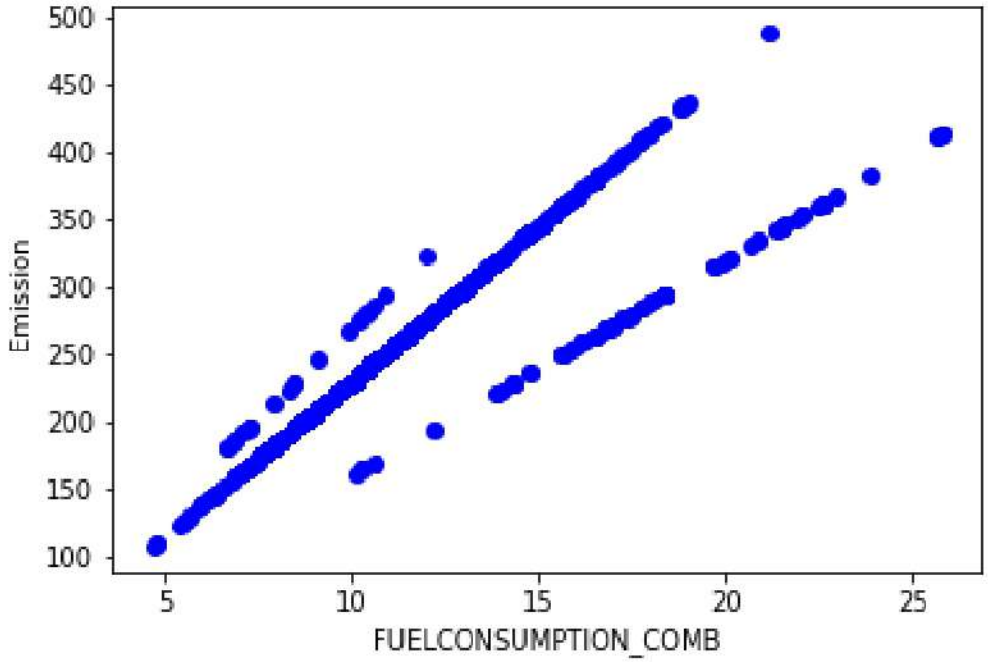
[15]:

	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
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267

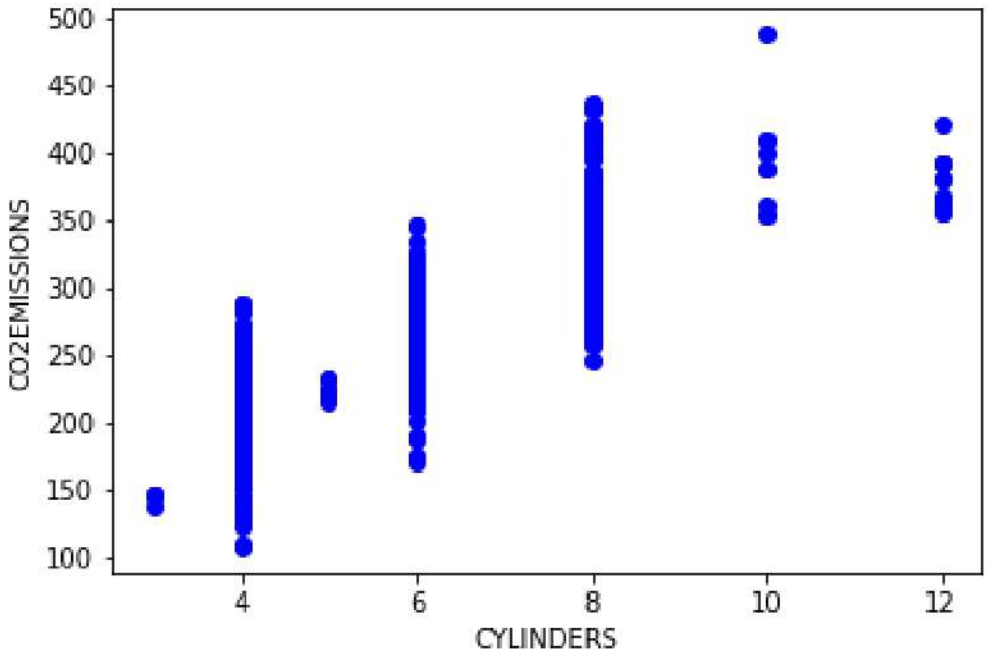
[16]:
```
viz=cdf[['ENGINESIZE','CYLINDERS','CO2EMISSIONS','FUELCONSUMPTION_COMB']]
viz.hist()
plt.show()
```



[17]:
```
plt.scatter(cdf.FUELCONSUMPTION_COMB,cdf.CO2EMISSIONS,color='blue')
plt.xlabel("FUELCONSUMPTION_COMB")
plt.ylabel("Emission")
plt.show()
```



[18]:
```
plt.scatter(cdf.CYLINDERS, cdf.CO2EMISSIONS, color='blue')
plt.xlabel("CYLINDERS")
plt.ylabel("CO2EMISSIONS")
plt.show()
```



[19]:
```
msk=np.random.rand(len(df))<0.8
train=cdf[msk]
test=cdf[~msk]
```

[20]:
```
train.head()
```

[20]:

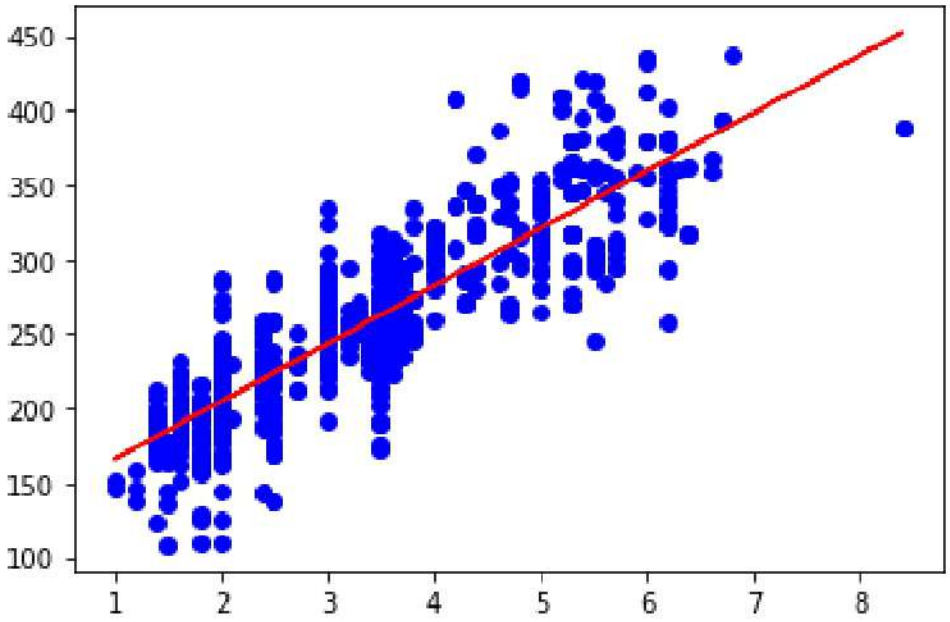
	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244

[21]:
```
from sklearn import linear_model
regressor= linear_model.LinearRegression()
train_x = np.asanyarray(train[['ENGINESIZE']])
train_y = np.asanyarray(train[['CO2EMISSIONS']])
```

```
regressor.fit(train_x, train_y)
print ('Coefficients: ', regressor.coef_)
print ('Intercept: ',regressor.intercept_)
```

```
Coefficients:  [[38.70857212]]
Intercept:  [127.13557218]
```

[22]:
```
plt.scatter(train.ENGINESIZE, train.CO2EMISSIONS, color='blue')
plt.plot(train_x, regressor.coef_[0][0]*train_x+ regressor.intercept_[0],'-r')
plt.show()
```



[30]:
```
from sklearn.metrics import r2_score
test_x = np.asanyarray(test[['ENGINESIZE']])
test_y = np.asanyarray(test[['CO2EMISSIONS']])
test_y_ = regressor.predict(test_x)
print("mean squared error: " %np.mean(np.absolute(test_y_-test_y)))
print("mean error:%.2f " % np.mean((test_y_-test_y)**2))
print("R-2 score:%.2f " % r2_score(test_y_,test_y))
```

```
mean squared error:
mean error:897.68
R-2 score:0.71
```

[31]:
```
from sklearn.metrics import r2_score

test_x = np.asanyarray(test[['ENGINESIZE']])
```

```
test_y = np.asanyarray(test[['CO2EMISSIONS']])
test_y_ = regressor.predict(test_x)

print("Mean absolute error: %.2f" % np.mean(np.absolute(test_y_ - test_y)))
print("Residual sum of squares (MSE): %.2f" % np.mean((test_y_ - test_y) ** 2))
print("R2-score: %.2f" % r2_score(test_y_ , test_y) )
```

```
Mean absolute error: 23.22
Residual sum of squares (MSE): 897.68
R2-score: 0.71
```

[ ]: