In [49]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pylab as pl
%matplotlib inline
```

In [11]:

```
ata = pd.read_csv(r"https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDe
```

In [12]:

```
data
```

Out[12]:

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	TRANSMISSIC
0	2014	ACURA	ILX	COMPACT	2.0	4	A
1	2014	ACURA	ILX	COMPACT	2.4	4	M
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	A
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	A
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	A
...	...	...	...	...	...	...	
1062	2014	VOLVO	XC60 AWD	SUV - SMALL	3.0	6	A
1063	2014	VOLVO	XC60 AWD	SUV - SMALL	3.2	6	A
1064	2014	VOLVO	XC70 AWD	SUV - SMALL	3.0	6	A
1065	2014	VOLVO	XC70 AWD	SUV - SMALL	3.2	6	A
1066	2014	VOLVO	XC90 AWD	SUV - STANDARD	3.2	6	A

1067 rows × 13 columns

In [13]:

```
data.isnull().sum()
```

Out[13]:

```
MODELYEAR                 0
MAKE                      0
MODEL                     0
VEHICLECLASS              0
ENGINESIZE                0
CYLINDERS                 0
TRANSMISSION              0
FUELTYPE                  0
FUELCONSUMPTION_CITY      0
FUELCONSUMPTION_HWY       0
FUELCONSUMPTION_COMB      0
FUELCONSUMPTION_COMB_MPG  0
CO2EMISSIONS              0
dtype: int64
```

In [14]:

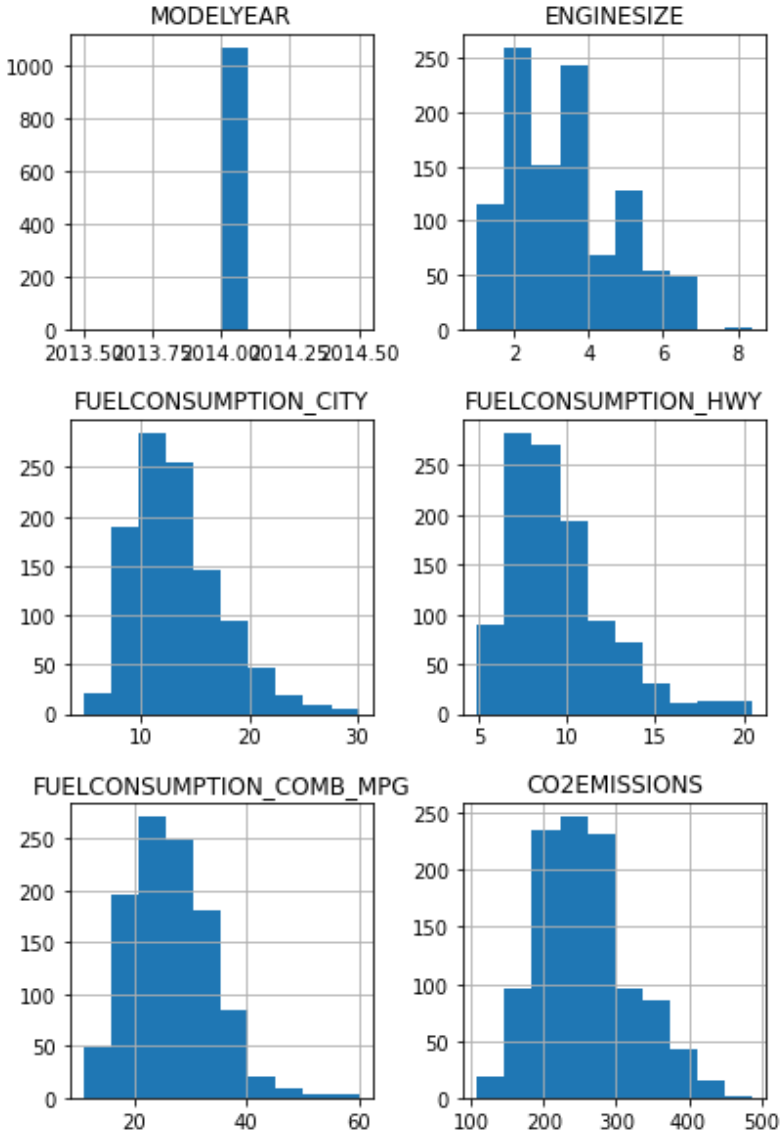
```
data.describe()
```

Out[14]:

	MODELYEAR	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_CITY	FUELCONSUMPTION_
count	1067.0	1067.000000	1067.000000	1067.000000	1067.0
mean	2014.0	3.346298	5.794752	13.296532	9.4
std	0.0	1.415895	1.797447	4.101253	2.7
min	2014.0	1.000000	3.000000	4.600000	4.9
25%	2014.0	2.000000	4.000000	10.250000	7.5
50%	2014.0	3.400000	6.000000	12.600000	8.8
75%	2014.0	4.300000	8.000000	15.550000	10.8
max	2014.0	8.400000	12.000000	30.200000	20.5

In [18]:

```
data.hist(figsize=(10,10))
plt.show()
```



In [23]:

```
car_data = data[["ENGINESIZE","CYLINDERS","FUELCONSUMPTION_COMB","CO2EMISSIONS"]]
```

In [26]:

```
car_data
```

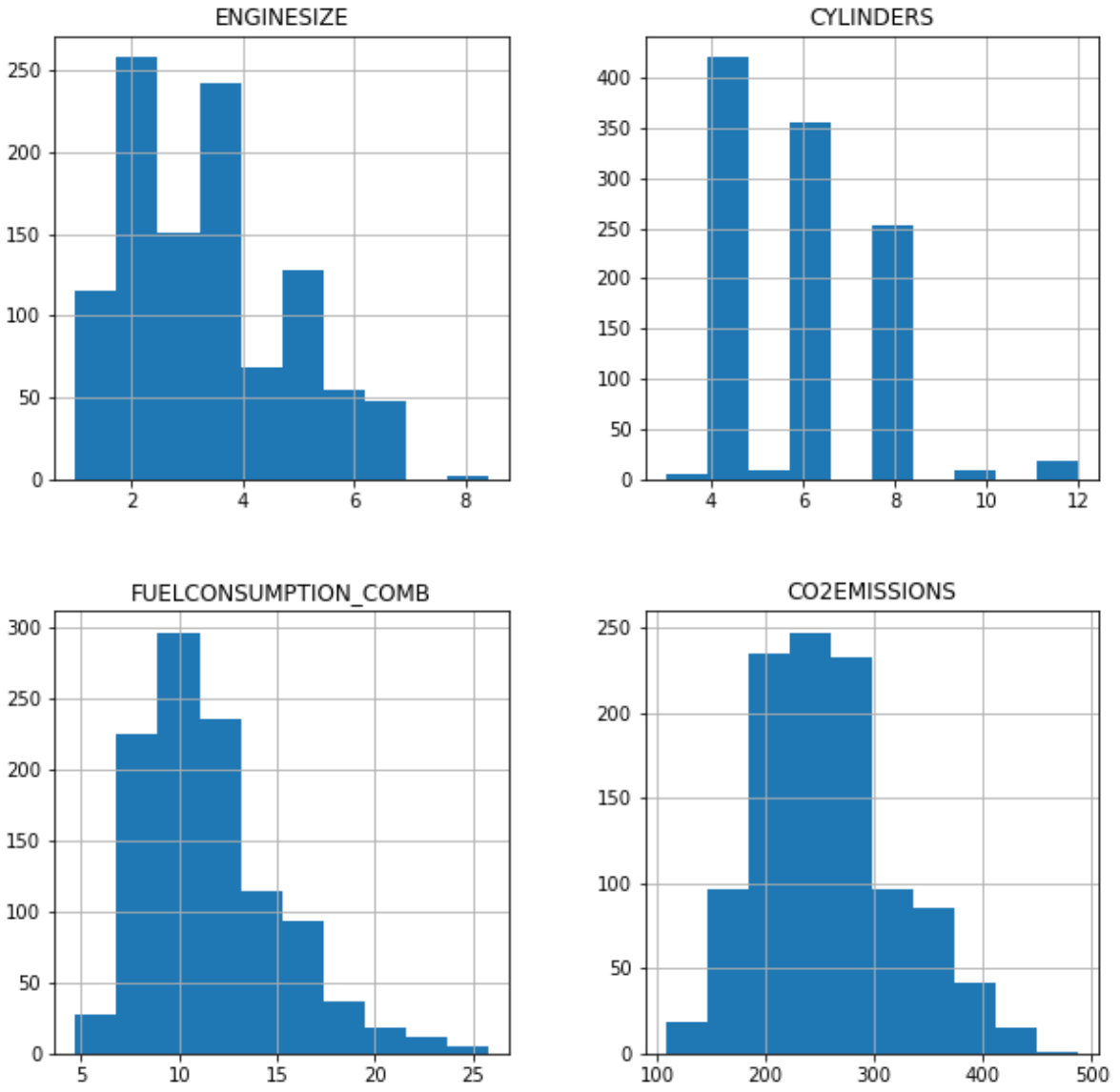
Out[26]:

	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
...	...	...	...	...
1062	3.0	6	11.8	271
1063	3.2	6	11.5	264
1064	3.0	6	11.8	271
1065	3.2	6	11.3	260
1066	3.2	6	12.8	294

1067 rows × 4 columns

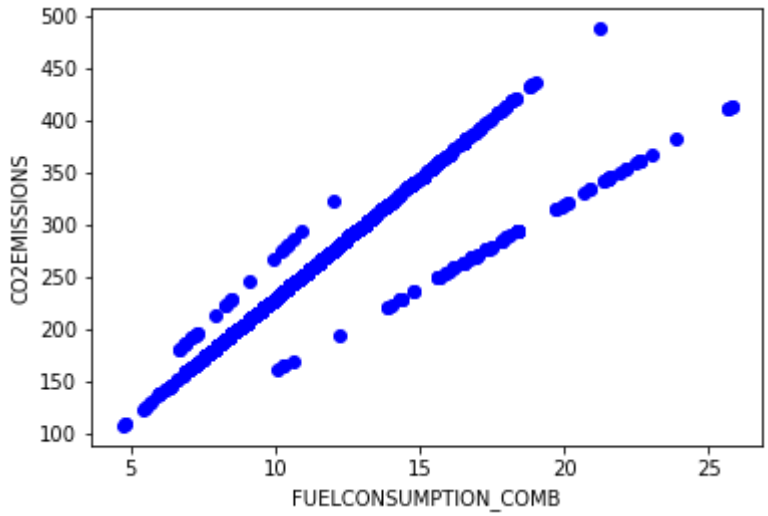
In [27]:

```
car_data.hist(figsize=(10,10))
plt.show()
```



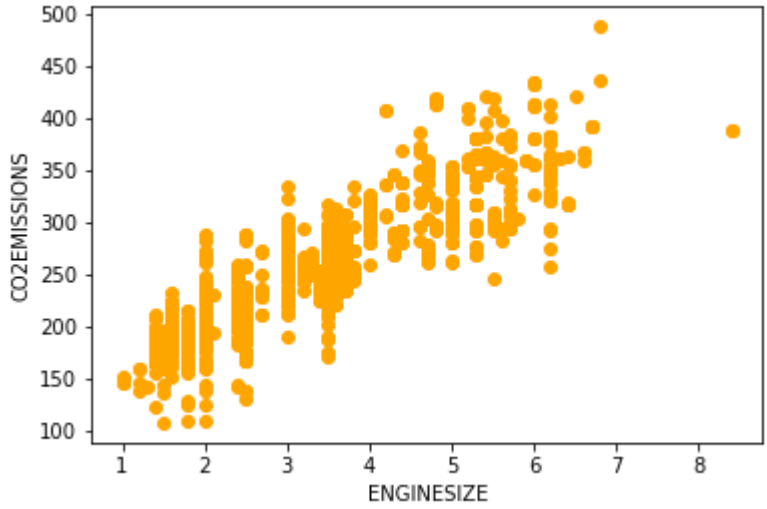
In [37]:

```
plt.scatter(car_data.FUELCONSUMPTION_COMB,car_data.CO2EMISSIONS,color='blue')
plt.xlabel('FUELCONSUMPTION_COMB')
plt.ylabel("CO2EMISSIONS")
plt.show()
```



In [47]:

```
plt.scatter(car_data.ENGINESIZE,car_data.CO2EMISSIONS,color='orange')
plt.xlabel("ENGINESIZE")
plt.ylabel("CO2EMISSIONS")
plt.show()
```



In [55]:

```
data_traine =np.random.rand(len(data))<0.8
train = car_data[data_traine]
test = car_data[~data_traine]
```

In [57]:

```
train.head(10)
```

Out[57]:

	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
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
13	5.9	12	15.6	359
14	4.7	8	14.7	338

In [56]:

```
test
```

Out[56]:

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
4	3.5	6	10.6	244
9	2.4	4	9.2	212
10	2.4	4	9.8	225
11	3.5	6	10.4	239
12	5.9	12	15.6	359
...	...	...	...	...
1047	3.6	6	10.8	248
1053	2.0	4	10.7	246
1055	3.0	6	10.4	281
1065	3.2	6	11.3	260
1066	3.2	6	12.8	294

211 rows × 4 columns

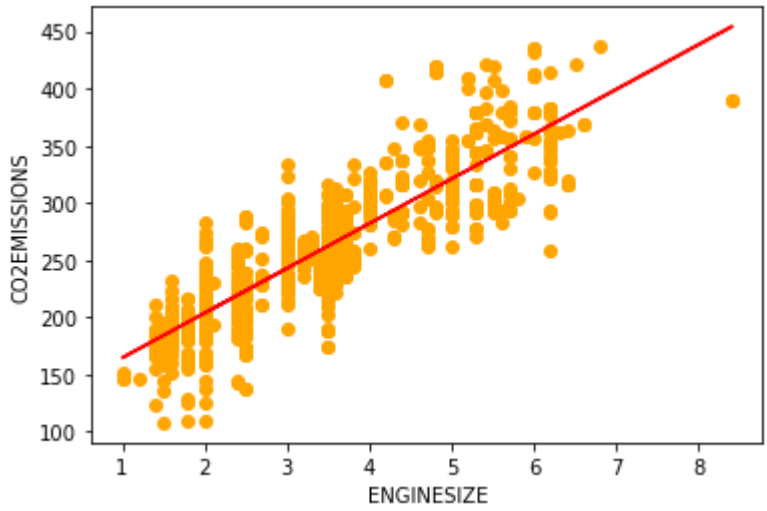
In [93]:

```
from sklearn import linear_model
regr = linear_model.LinearRegression()
train_X = np.asanyarray(train[['ENGINESIZE']])
train_Y=np.asanyarray(train[['CO2EMISSIONS']])
regr.fit(train_X,train_Y)
#find interpet,coff
print('interpet :',regr.intercept_,'\ncofficient :',regr.coef_)
```

```
interpet : [125.52630658]
cofficient : [[39.12432262]]
```

In [90]:

```
#plot fering line
plt.scatter(train.ENGINESIZE,train.CO2EMISSIONS,color = 'orange')
plt.plot(train_X,regr.coef_[0][0]*train_X+regr.intercept_,'-r')
plt.xlabel("ENGINESIZE")
plt.ylabel("CO2EMISSIONS")
plt.show()
```



In [91]:

```
from sklearn.metrics import r2_score
test_X = np.asanyarray(test[['ENGINESIZE']])
test_Y = np.asanyarray(test[['CO2EMISSIONS']])
test_Y_ = regr.predict(test_X)
print("mean absolute error %2.f" % np.mean(np.absolute(test_Y_- test_Y)))
print("mean square error %2.f" %np.mean((test_Y_- test_Y)**2))
print("r2 score %2.f" % r2_score(test_Y, test_Y_))
```

```
mean absolute error 21
mean square error 820
r2 score  1
```

In [86]:

```
from sklearn.metrics import r2_score

test_x = np.asanyarray(test[['ENGINESIZE']])
test_y = np.asanyarray(test[['CO2EMISSIONS']])
test_y_ = regr.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: 21.29
Residual sum of squares (MSE): 819.92
R2-score: 0.80
```

In [ ]:

In [ ]: