In [1]:
```python
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
data=pd.read_csv(r"C:\Users\user\Desktop\DINESH\C10_air\madrid_2012.csv")
data
```

Out[2]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
0	2012-09-01 01:00:00	NaN	0.2	NaN	NaN	7.0	18.0	NaN	NaN	NaN	2.0	NaN	NaN	28
1	2012-09-01 01:00:00	0.3	0.3	0.7	NaN	3.0	18.0	55.0	10.0	9.0	1.0	NaN	2.4	28
2	2012-09-01 01:00:00	0.4	NaN	0.7	NaN	2.0	10.0	NaN	NaN	NaN	NaN	NaN	1.5	28
3	2012-09-01 01:00:00	NaN	0.2	NaN	NaN	1.0	6.0	50.0	NaN	NaN	NaN	NaN	NaN	28
4	2012-09-01 01:00:00	NaN	NaN	NaN	NaN	1.0	13.0	54.0	NaN	NaN	3.0	NaN	NaN	28
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210715	2012-03-01 00:00:00	NaN	0.6	NaN	NaN	37.0	84.0	14.0	NaN	NaN	NaN	NaN	NaN	28
210716	2012-03-01 00:00:00	NaN	0.4	NaN	NaN	5.0	76.0	NaN	17.0	NaN	7.0	NaN	NaN	28
210717	2012-03-01 00:00:00	NaN	NaN	NaN	0.34	3.0	41.0	24.0	NaN	NaN	NaN	1.34	NaN	28
210718	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	44.0	36.0	NaN	NaN	NaN	NaN	NaN	28
210719	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	56.0	40.0	18.0	NaN	NaN	NaN	NaN	28

210720 rows × 14 columns

In [3]:
```
data.head(10)
```

Out[3]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	station
0	2012-09-01 01:00:00	NaN	0.2	NaN	NaN	7.0	18.0	NaN	NaN	NaN	2.0	NaN	NaN	28079004
1	2012-09-01 01:00:00	0.3	0.3	0.7	NaN	3.0	18.0	55.0	10.0	9.0	1.0	NaN	2.4	28079008
2	2012-09-01 01:00:00	0.4	NaN	0.7	NaN	2.0	10.0	NaN	NaN	NaN	NaN	NaN	1.5	28079011
3	2012-09-01 01:00:00	NaN	0.2	NaN	NaN	1.0	6.0	50.0	NaN	NaN	NaN	NaN	NaN	28079016
4	2012-09-01 01:00:00	NaN	NaN	NaN	NaN	1.0	13.0	54.0	NaN	NaN	3.0	NaN	NaN	28079017
5	2012-09-01 01:00:00	0.2	0.2	1.0	NaN	1.0	9.0	57.0	14.0	NaN	1.0	NaN	0.2	28079018
6	2012-09-01 01:00:00	0.4	0.2	0.8	0.24	1.0	7.0	57.0	11.0	7.0	2.0	1.33	0.6	28079024
7	2012-09-01 01:00:00	NaN	NaN	NaN	0.11	1.0	2.0	65.0	NaN	NaN	NaN	1.18	NaN	28079027
8	2012-09-01 01:00:00	NaN	0.2	NaN	NaN	6.0	14.0	57.0	NaN	NaN	2.0	NaN	NaN	28079035
9	2012-09-01 01:00:00	NaN	0.2	NaN	NaN	1.0	7.0	NaN	13.0	NaN	1.0	NaN	NaN	28079036

In [4]:
```
data.tail(20)
```

Out[4]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
210700	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	7.0	68.0	20.0	NaN	NaN	3.0	NaN	NaN	28
210701	2012-03-01 00:00:00	0.2	0.5	0.9	NaN	5.0	55.0	33.0	20.0	NaN	5.0	NaN	1.1	28
210702	2012-03-01 00:00:00	0.6	0.3	0.5	0.09	1.0	23.0	61.0	18.0	16.0	3.0	1.11	1.2	28
210703	2012-03-01 00:00:00	NaN	NaN	NaN	0.19	6.0	69.0	28.0	NaN	NaN	NaN	1.32	NaN	28
210704	2012-03-01 00:00:00	NaN	0.5	NaN	NaN	6.0	56.0	33.0	NaN	NaN	8.0	NaN	NaN	28
210705	2012-03-01 00:00:00	NaN	0.4	NaN	NaN	3.0	70.0	NaN	21.0	NaN	13.0	NaN	NaN	28
210706	2012-03-01 00:00:00	0.2	NaN	0.2	NaN	10.0	48.0	NaN	21.0	15.0	7.0	NaN	2.6	28
210707	2012-03-01 00:00:00	NaN	0.4	NaN	NaN	10.0	71.0	27.0	NaN	NaN	NaN	NaN	NaN	28
210708	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	3.0	57.0	NaN	22.0	NaN	8.0	NaN	NaN	28
210709	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	1.0	56.0	NaN	20.0	16.0	NaN	NaN	NaN	28
210710	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	5.0	55.0	NaN	21.0	15.0	NaN	NaN	NaN	28
210711	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	1.0	37.0	35.0	NaN	NaN	NaN	NaN	NaN	28
210712	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	23.0	69.0	NaN	25.0	9.0	NaN	NaN	NaN	28
210713	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	1.0	51.0	39.0	NaN	NaN	NaN	NaN	NaN	28
210714	2012-03-01 00:00:00	0.8	NaN	0.5	0.16	51.0	104.0	NaN	23.0	NaN	NaN	1.48	2.9	28
210715	2012-03-01 00:00:00	NaN	0.6	NaN	NaN	37.0	84.0	14.0	NaN	NaN	NaN	NaN	NaN	28
210716	2012-03-01 00:00:00	NaN	0.4	NaN	NaN	5.0	76.0	NaN	17.0	NaN	7.0	NaN	NaN	28

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
210717	2012-03-01 00:00:00	NaN	NaN	NaN	0.34	3.0	41.0	24.0	NaN	NaN	NaN	1.34	NaN	28
210718	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	44.0	36.0	NaN	NaN	NaN	NaN	NaN	28
210719	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	56.0	40.0	18.0	NaN	NaN	NaN	NaN	28

In [5]:
```
data.describe()
```

Out[5]:

	BEN	CO	EBE	NMHC	NO	NO_2	
count	51511.000000	87097.000000	51482.000000	30736.000000	209871.000000	209872.000000	1:
mean	0.829037	0.355027	0.951987	0.187244	24.743719	38.653698	
std	0.889463	0.250771	0.826109	0.098950	49.852175	29.011524	
min	0.000000	0.100000	0.000000	0.000000	0.000000	1.000000	
25%	0.200000	0.200000	0.500000	0.120000	2.000000	17.000000	
50%	0.500000	0.300000	0.900000	0.170000	7.000000	32.000000	
75%	1.100000	0.400000	1.000000	0.240000	23.000000	54.000000	
max	13.400000	4.400000	25.200001	2.210000	933.000000	353.000000	

In [6]:
```
np.shape(data)
```

Out[6]: (210720, 14)

In [7]:
```
np.size(data)
```

Out[7]: 2950080

In [8]:
```
data.isna()
```

Out[8]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL
0	False	True	False	True	True	False	False	True	True	True	False	True	True
1	False	False	False	False	True	False	False	False	False	False	False	True	False
2	False	False	True	False	True	False	False	True	True	True	True	True	False
3	False	True	False	True	True	False	False	False	True	True	True	True	True
4	False	True	True	True	True	False	False	False	True	True	False	True	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...
210715	False	True	False	True	True	False	False	False	True	True	True	True	True
210716	False	True	False	True	True	False	False	True	False	True	False	True	True
210717	False	True	True	True	False	False	False	False	True	True	True	False	True
210718	False	True	True	True	True	False	False	False	True	True	True	True	True
210719	False	True	True	True	True	False	False	False	False	True	True	True	True

210720 rows × 14 columns

In [9]:
```python
data.dropna()
```

Out[9]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	s
6	2012-09-01 01:00:00	0.4	0.2	0.8	0.24	1.0	7.0	57.0	11.0	7.0	2.0	1.33	0.6	280
30	2012-09-01 02:00:00	0.4	0.2	0.7	0.24	1.0	5.0	55.0	5.0	5.0	2.0	1.33	0.5	280
54	2012-09-01 03:00:00	0.4	0.2	0.7	0.24	1.0	4.0	56.0	6.0	4.0	2.0	1.33	0.5	280
78	2012-09-01 04:00:00	0.3	0.2	0.7	0.25	1.0	5.0	54.0	6.0	5.0	2.0	1.34	0.4	280
102	2012-09-01 05:00:00	0.4	0.2	0.7	0.24	1.0	3.0	53.0	8.0	5.0	2.0	1.33	0.5	280
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210654	2012-02-29 22:00:00	0.6	0.3	0.5	0.09	1.0	35.0	57.0	25.0	21.0	3.0	1.12	2.3	280
210673	2012-02-29 23:00:00	2.0	0.4	2.4	0.21	16.0	79.0	20.0	37.0	25.0	12.0	1.33	6.2	280
210678	2012-02-29 23:00:00	0.7	0.3	0.6	0.09	1.0	27.0	63.0	22.0	18.0	3.0	1.11	1.9	280
210697	2012-03-01 00:00:00	1.5	0.4	1.7	0.21	16.0	79.0	17.0	28.0	21.0	11.0	1.34	4.9	280
210702	2012-03-01 00:00:00	0.6	0.3	0.5	0.09	1.0	23.0	61.0	18.0	16.0	3.0	1.11	1.2	280

10916 rows × 14 columns

In [10]:
```python
data.columns
```

Out[10]:
```
Index(['date', 'BEN', 'CO', 'EBE', 'NMHC', 'NO', 'NO_2', 'O_3', 'PM10', 'PM2
5',
       'SO_2', 'TCH', 'TOL', 'station'],
      dtype='object')
```

In [11]:
```python
sd=data[['BEN','CO', 'EBE', 'NMHC', 'NO_2']]
```

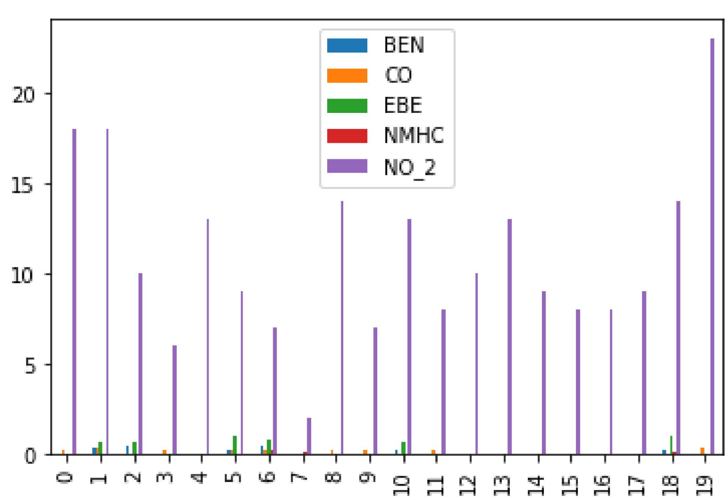
In [12]:
```
dd=sd.head(20)
dd
```

Out[12]:

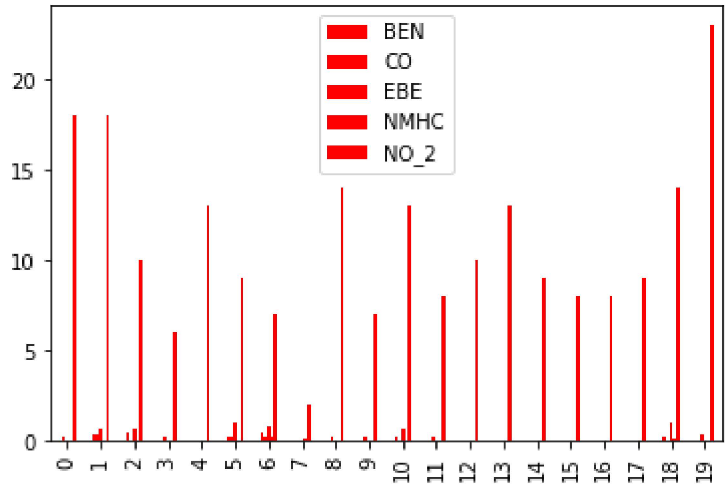
	BEN	CO	EBE	NMHC	NO_2
0	NaN	0.2	NaN	NaN	18.0
1	0.3	0.3	0.7	NaN	18.0
2	0.4	NaN	0.7	NaN	10.0
3	NaN	0.2	NaN	NaN	6.0
4	NaN	NaN	NaN	NaN	13.0
5	0.2	0.2	1.0	NaN	9.0
6	0.4	0.2	0.8	0.24	7.0
7	NaN	NaN	NaN	0.11	2.0
8	NaN	0.2	NaN	NaN	14.0
9	NaN	0.2	NaN	NaN	7.0
10	0.2	NaN	0.7	NaN	13.0
11	NaN	0.2	NaN	NaN	8.0
12	NaN	NaN	NaN	NaN	10.0
13	NaN	NaN	NaN	NaN	13.0
14	NaN	NaN	NaN	NaN	9.0
15	NaN	NaN	NaN	NaN	8.0
16	NaN	NaN	NaN	NaN	8.0
17	NaN	NaN	NaN	NaN	9.0
18	0.2	NaN	1.0	0.09	14.0
19	NaN	0.3	NaN	NaN	23.0

In [13]:
```
dd.plot.bar()
```

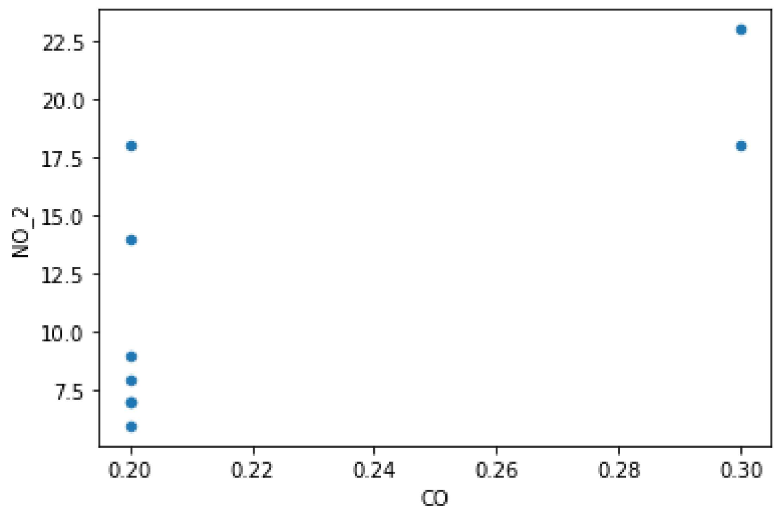
Out[13]: <AxesSubplot:>



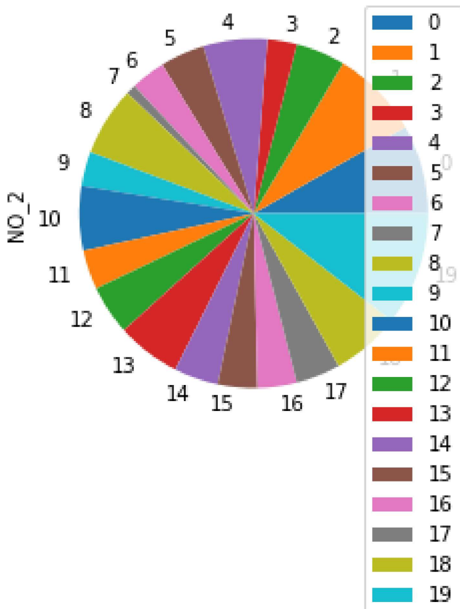
In [14]:
```
dd.plot.bar(color='r')
```
Out[14]: <AxesSubplot:>



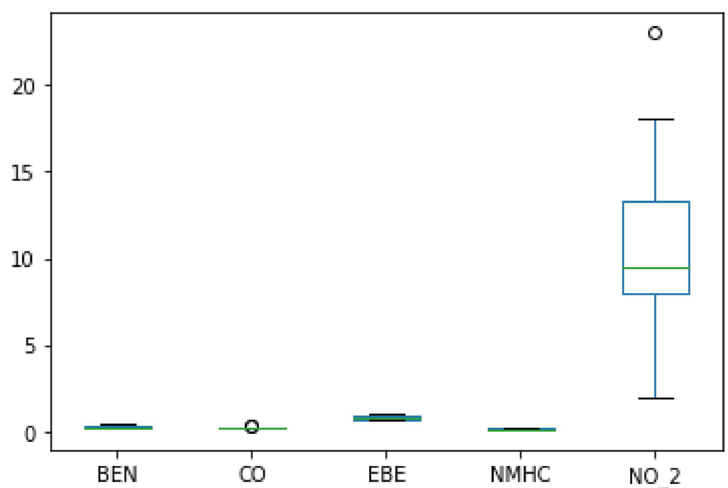
In [15]:
```
dd.plot.scatter(x='CO',y='NO_2')
```
Out[15]: <AxesSubplot:xlabel='CO', ylabel='NO_2'>



In [16]:
```
dd.plot.pie(y='NO_2')
```
Out[16]: <AxesSubplot:ylabel='NO_2'>

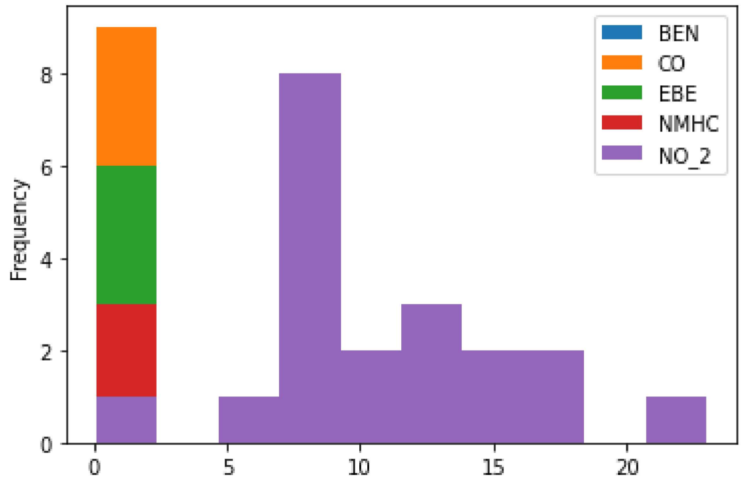


In [17]:
```
dd.plot.box()
```
Out[17]: <AxesSubplot:>



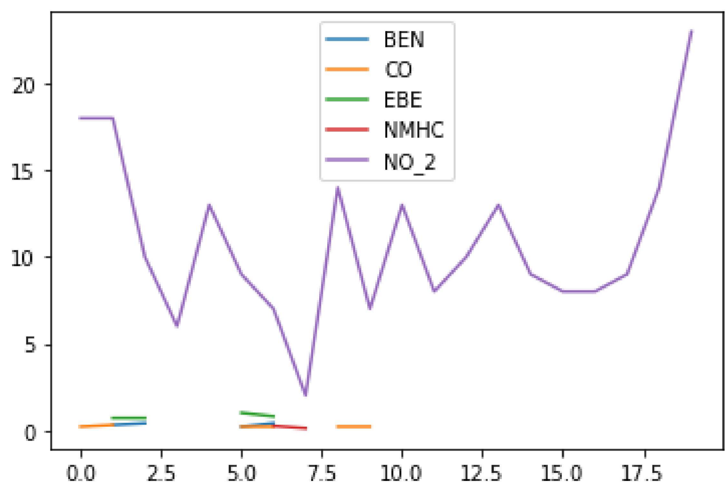
In [18]:
```
dd.plot.hist()
```

Out[18]: <AxesSubplot:ylabel='Frequency'>

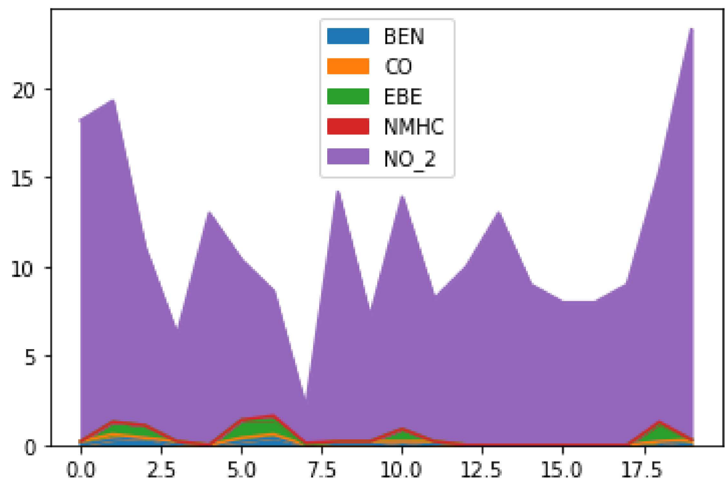


In [19]:
```
dd.plot.line()
```

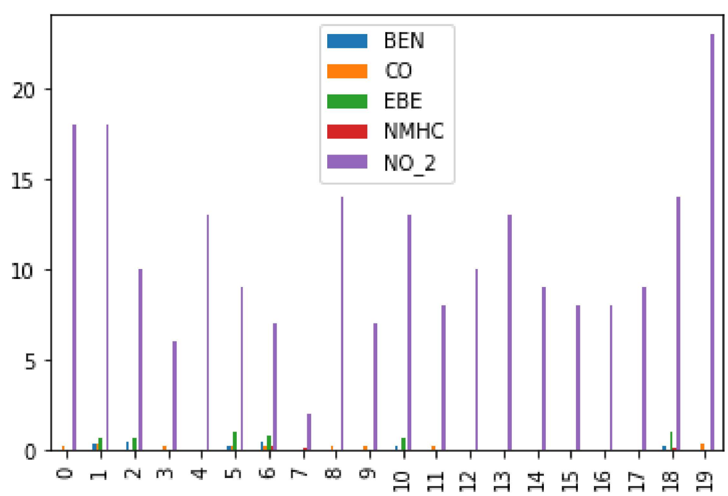
Out[19]: <AxesSubplot:>



In [20]:
```
dd.plot.area()
```
Out[20]: <AxesSubplot:>

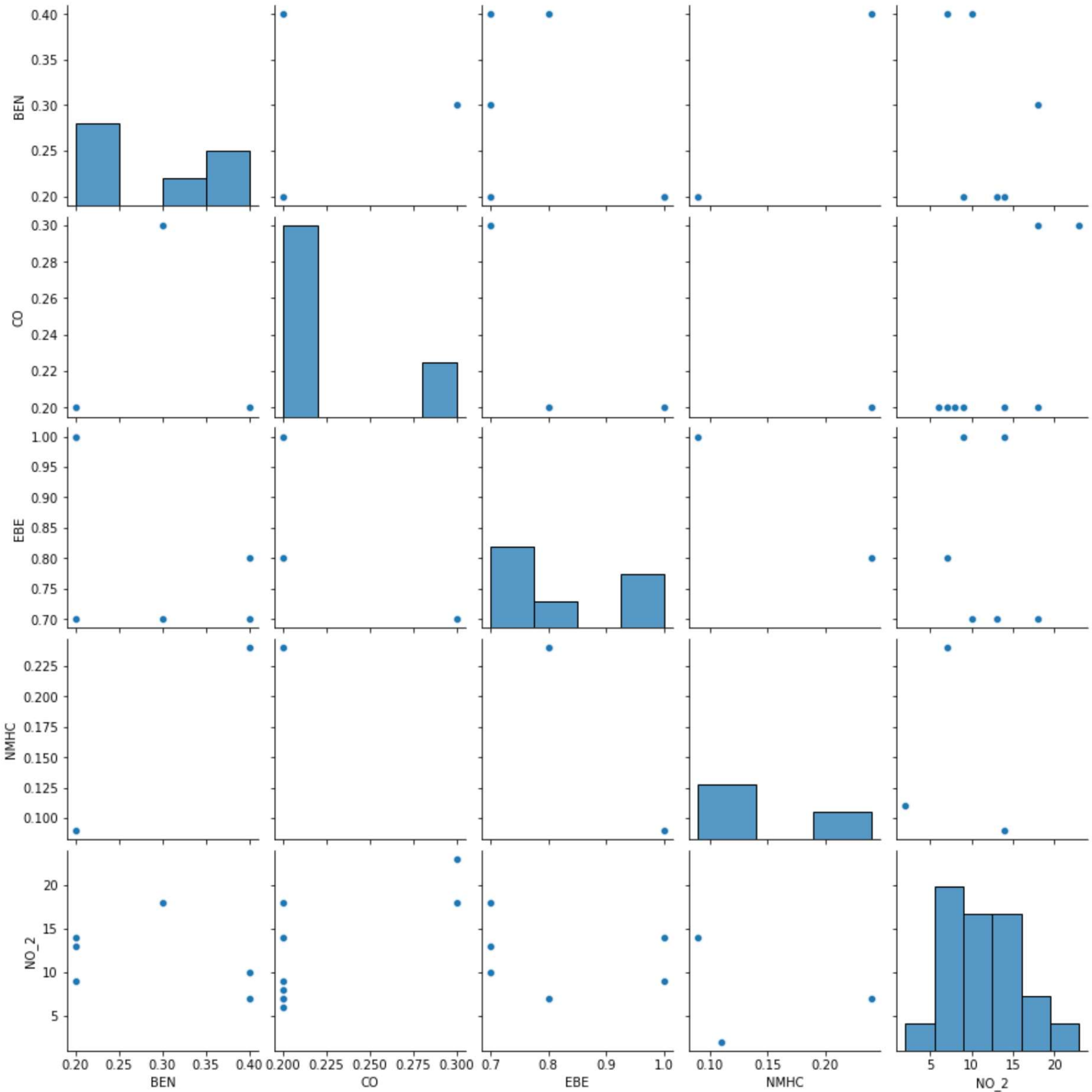


In [21]:
```
dd.plot.bar()
```
Out[21]: <AxesSubplot:>



In [22]:
```
sns.pairplot(dd)
```

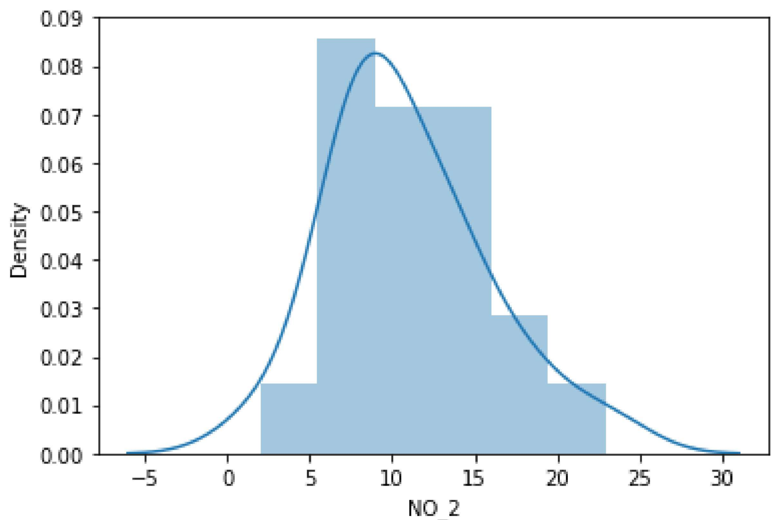
Out[22]: <seaborn.axisgrid.PairGrid at 0x26982e7a820>



In [23]:
```
sns.distplot(dd['NO_2'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[23]:
```
<AxesSubplot:xlabel='NO_2', ylabel='Density'>
```



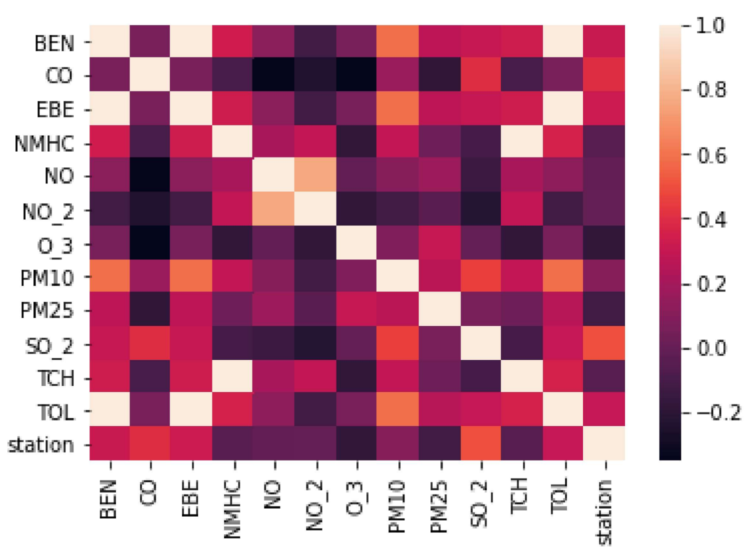
In [24]:
```
ds=data.fillna(20)
```

In [25]:
```
ssd=ds.head(20)
```

In [26]:
```
sd1=ssd[['BEN','CO', 'EBE', 'NMHC', 'NO_2']]
```

In [27]:
```
sns.heatmap(ssd.corr())
```

Out[27]:
```
<AxesSubplot:>
```



In [28]:
```
x= ssd[['BEN','CO', 'EBE','NMHC', 'NO_2']]
y=ssd['station']
```

In [29]:
```
from sklearn .model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [30]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[30]: LinearRegression()

In [31]:
```
print(lr.intercept_)
```

28078998.133279286

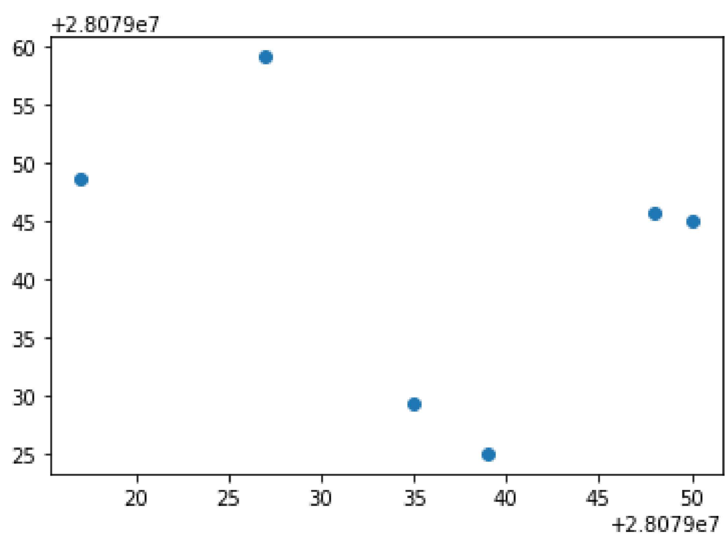
In [32]:
```
coeff= pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[32]:

	Co-efficient
BEN	-36.798864
CO	1.011817
EBE	38.771170
NMHC	-0.918907
NO_2	0.707346

In [33]:
```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[33]: <matplotlib.collections.PathCollection at 0x2698743d760>



In [34]:
```
print(lr.score(x_test,y_test))
```

```
-1.8895092983378525
```

In [35]:
```
lr.score(x_test,y_test)
```

Out[35]:
```
-1.8895092983378525
```

In [36]:
```
lr.score(x_train,y_train)
```

Out[36]:
```
0.6205569518122
```

In [37]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [38]:
```
dr=Ridge(alpha=10)
dr.fit(x_train,y_train)
```

Out[38]:
```
Ridge(alpha=10)
```

In [39]:
```
dr.score(x_test,y_test)
```

Out[39]:
```
-2.1586744939282436
```

In [40]:
```
dr.score(x_train,y_train)
```

Out[40]:
```
0.5524442983764888
```

In [41]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[41]:
```
Lasso(alpha=10)
```

In [42]:
```
la.score(x_test,y_test)
```

Out[42]:
```
-1.4879303595429452
```

In [43]:
```
la.score(x_train,y_train)
```

Out[43]:
```
0.511095433719627
```

ElasticNet

In [44]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[44]:
```
ElasticNet()
```

In [45]:
```
print(en.coef_)
```

```
[ 0.          0.99937341  0.99675915 -1.07774239  0.69400721]
```

In [46]:
```
print(en.intercept_)
```

```
28079021.044698644
```

In [47]:
```
prediction=en.predict(x_test)
```

In [48]:
```
print(en.score(x_test,y_test))
```

```
-2.135991381769106
```

In [49]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [50]:
```
from sklearn.linear_model import LogisticRegression
```

In [51]:
```
feature_matrix = ssd[['BEN','CO', 'EBE','NMHC', 'NO_2']]
target_vector=ssd['station']
```

In [52]:
```
feature_matrix.shape
```

Out[52]:
```
(20, 5)
```

In [53]:
```
target_vector.shape
```

Out[53]:
```
(20,)
```

In [54]:
```
from sklearn.preprocessing import StandardScaler
```

In [55]:
```
fs=StandardScaler().fit_transform(feature_matrix)
```

In [56]:
```
logr= LogisticRegression()
logr.fit(fs,target_vector)
```

Out[56]:
```
LogisticRegression()
```

In [57]:
```
observation =[[1.2,2.3,3.3,4.3,5.3]]
```

In [58]:
```
prediction=logr.predict(observation)
print(prediction)
```

```
[28079056]
```

In [59]:
```
logr.classes_
```

Out[59]:
```
array([28079004, 28079008, 28079011, 28079016, 28079017, 28079018,
       28079024, 28079027, 28079035, 28079036, 28079038, 28079039,
       28079040, 28079047, 28079048, 28079049, 28079050, 28079054,
       28079055, 28079056], dtype=int64)
```

In [60]:
```
logr.predict_proba(observation)[0][0]
```

Out[60]:
```
0.05460453824864577
```

In [61]:
```
ged=data[['BEN','CO','EBE','NMHC','NO_2','O_3','PM10','SO_2','TCH','TOL','stati
```

In [62]:
```
d=ged.fillna(20)
```

In [63]:
```
dg=d.head(100)
```

In [64]:
```
x=dg[['BEN','CO','EBE','NMHC','NO_2','O_3','PM10','SO_2','TCH','TOL']]
y=dg['station']
```

In [65]:
```
print(len(x))
print(len(y))
```

```
100
100
```

In [66]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [67]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[67]:
```
RandomForestClassifier()
```

In [68]:
```
paramets = {'max_depth':[1,2,3,4,5,6,7],
              'min_samples_leaf':[5,10,15,20,25,30,35],
              'n_estimators':[10,20,30,40,50,60,70]}
```

In [69]:
```
from sklearn.model_selection import GridSearchCV
grid_search= GridSearchCV(estimator = rfc,param_grid=paramets,cv=2,scoring="acc
grid_search.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
666: UserWarning: The least populated class in y has only 1 members, which is
less than n_splits=2.
  warnings.warn(("The least populated class in y has only %d"
```

Out[69]:
```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5, 6, 7],
                         'min_samples_leaf': [5, 10, 15, 20, 25, 30, 35],
                         'n_estimators': [10, 20, 30, 40, 50, 60, 70]},
             scoring='accuracy')
```

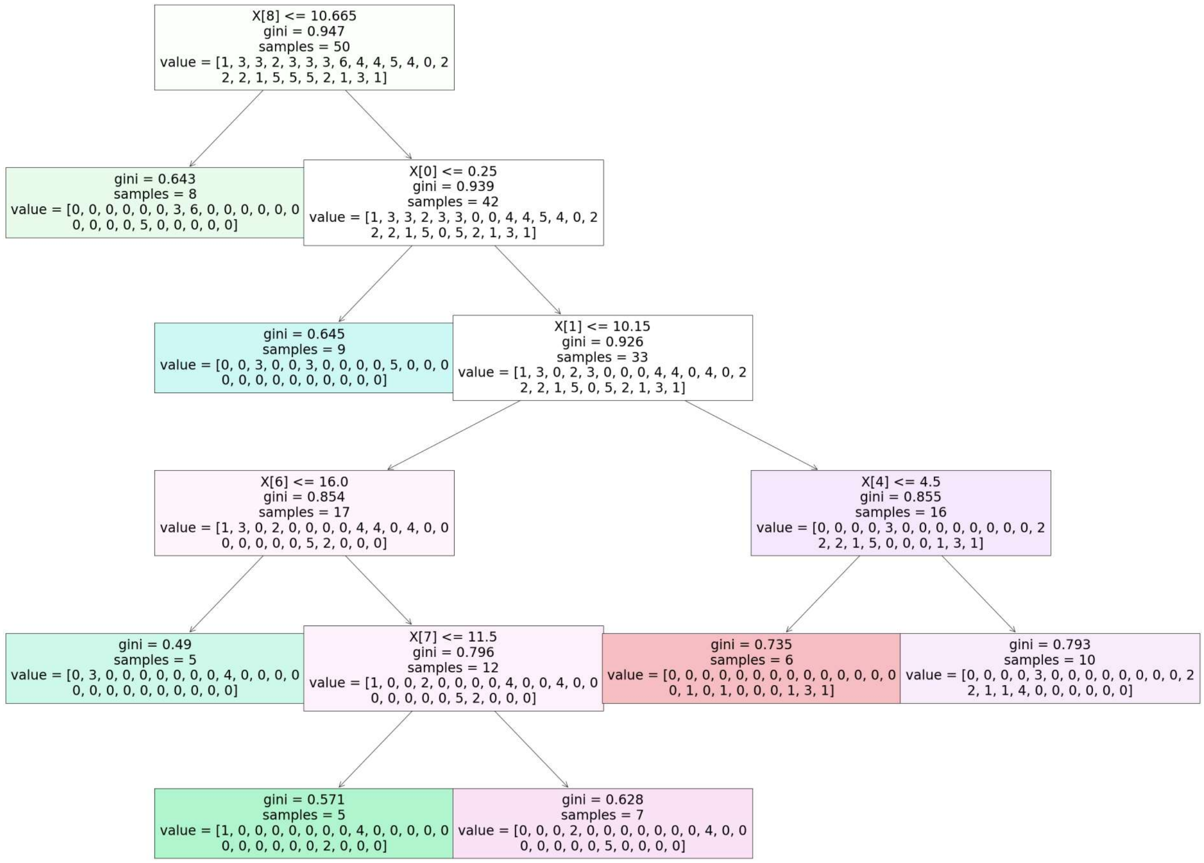
In [70]:
```
grid_search.best_score_
```

Out[70]:
```
0.4857142857142857
```

In [71]:
```
rfc_best=grid_search.best_estimator_
```

In [72]:
```python
from sklearn.tree import plot_tree
plt.figure(figsize=(50,40))
plot_tree(rfc_best.estimators_[5],filled=True)
```

Out[72]:
```
[Text(697.5, 1993.2, 'X[8] <= 10.665\ngini = 0.947\nsamples = 50\nvalue = [1, 3, 3, 2, 3, 3, 3, 6, 4, 4, 5, 4, 0, 2\n2, 2, 1, 5, 5, 5, 2, 1, 3, 1]'),
 Text(348.75, 1630.8000000000002, 'gini = 0.643\nsamples = 8\nvalue = [0, 0, 0, 0, 0, 0, 3, 6, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 5, 0, 0, 0, 0, 0]'),
 Text(1046.25, 1630.8000000000002, 'X[0] <= 0.25\ngini = 0.939\nsamples = 42\nvalue = [1, 3, 3, 2, 3, 3, 0, 0, 4, 4, 5, 4, 0, 2\n2, 2, 1, 5, 0, 5, 2, 1, 3, 1]'),
 Text(697.5, 1268.4, 'gini = 0.645\nsamples = 9\nvalue = [0, 0, 3, 0, 0, 3, 0, 0, 0, 0, 5, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1395.0, 1268.4, 'X[1] <= 10.15\ngini = 0.926\nsamples = 33\nvalue = [1, 3, 0, 2, 3, 0, 0, 0, 4, 4, 0, 4, 0, 2\n2, 2, 1, 5, 0, 5, 2, 1, 3, 1]'),
 Text(697.5, 906.0, 'X[6] <= 16.0\ngini = 0.854\nsamples = 17\nvalue = [1, 3, 0, 2, 0, 0, 0, 0, 4, 4, 0, 4, 0, 0\n0, 0, 0, 0, 0, 5, 2, 0, 0, 0]'),
 Text(348.75, 543.5999999999999, 'gini = 0.49\nsamples = 5\nvalue = [0, 3, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1046.25, 543.5999999999999, 'X[7] <= 11.5\ngini = 0.796\nsamples = 12\nvalue = [1, 0, 0, 2, 0, 0, 0, 0, 4, 0, 0, 4, 0, 0\n0, 0, 0, 0, 0, 5, 2, 0, 0, 0]'),
 Text(697.5, 181.19999999999982, 'gini = 0.571\nsamples = 5\nvalue = [1, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 2, 0, 0, 0]'),
 Text(1395.0, 181.19999999999982, 'gini = 0.628\nsamples = 7\nvalue = [0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0\n0, 0, 0, 0, 0, 5, 0, 0, 0, 0]'),
 Text(2092.5, 906.0, 'X[4] <= 4.5\ngini = 0.855\nsamples = 16\nvalue = [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 2\n2, 2, 1, 5, 0, 0, 0, 1, 3, 1]'),
 Text(1743.75, 543.5999999999999, 'gini = 0.735\nsamples = 6\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 1, 0, 1, 0, 0, 0, 1, 3, 1]'),
 Text(2441.25, 543.5999999999999, 'gini = 0.793\nsamples = 10\nvalue = [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 2\n2, 1, 1, 4, 0, 0, 0, 0, 0, 0]')]
```



Conclusion : ElasticNet() 28079021.044698644 HIGH RANGE

In [ ]: