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

In [2]:
```
data=pd.read_csv(r"C:\Users\user\Desktop\DINESH\C10_air\madrid_2013.csv")
data
```

Out[2]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
0	2013-11-01 01:00:00	NaN	0.6	NaN	NaN	135.0	74.0	NaN	NaN	NaN	7.0	NaN	NaN	2
1	2013-11-01 01:00:00	1.5	0.5	1.3	NaN	71.0	83.0	2.0	23.0	16.0	12.0	NaN	8.3	2
2	2013-11-01 01:00:00	3.9	NaN	2.8	NaN	49.0	70.0	NaN	NaN	NaN	NaN	NaN	9.0	2
3	2013-11-01 01:00:00	NaN	0.5	NaN	NaN	82.0	87.0	3.0	NaN	NaN	NaN	NaN	NaN	2
4	2013-11-01 01:00:00	NaN	NaN	NaN	NaN	242.0	111.0	2.0	NaN	NaN	12.0	NaN	NaN	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
209875	2013-03-01 00:00:00	NaN	0.4	NaN	NaN	8.0	39.0	52.0	NaN	NaN	NaN	NaN	NaN	2
209876	2013-03-01 00:00:00	NaN	0.4	NaN	NaN	1.0	11.0	NaN	6.0	NaN	2.0	NaN	NaN	2
209877	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	4.0	75.0	NaN	NaN	NaN	NaN	NaN	2
209878	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	11.0	52.0	NaN	NaN	NaN	NaN	NaN	2
209879	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	1.0	10.0	75.0	3.0	NaN	NaN	NaN	NaN	2

209880 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	stati
0	2013-11-01 01:00:00	NaN	0.6	NaN	NaN	135.0	74.0	NaN	NaN	NaN	7.0	NaN	NaN	280790
1	2013-11-01 01:00:00	1.5	0.5	1.3	NaN	71.0	83.0	2.0	23.0	16.0	12.0	NaN	8.3	280790
2	2013-11-01 01:00:00	3.9	NaN	2.8	NaN	49.0	70.0	NaN	NaN	NaN	NaN	NaN	9.0	280790
3	2013-11-01 01:00:00	NaN	0.5	NaN	NaN	82.0	87.0	3.0	NaN	NaN	NaN	NaN	NaN	280790
4	2013-11-01 01:00:00	NaN	NaN	NaN	NaN	242.0	111.0	2.0	NaN	NaN	12.0	NaN	NaN	280790
5	2013-11-01 01:00:00	1.0	0.6	0.8	NaN	70.0	70.0	2.0	24.0	NaN	6.0	NaN	5.2	280790
6	2013-11-01 01:00:00	NaN	0.4	NaN	0.29	51.0	80.0	5.0	23.0	14.0	4.0	1.44	NaN	280790
7	2013-11-01 01:00:00	NaN	NaN	NaN	0.23	29.0	60.0	4.0	NaN	NaN	NaN	1.51	NaN	280790
8	2013-11-01 01:00:00	NaN	1.0	NaN	NaN	165.0	107.0	2.0	NaN	NaN	11.0	NaN	NaN	280790
9	2013-11-01 01:00:00	NaN	0.6	NaN	NaN	63.0	93.0	NaN	11.0	NaN	8.0	NaN	NaN	280790

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

Out[4]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	s
209860	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	1.0	13.0	69.0	NaN	NaN	3.0	NaN	NaN	280
209861	2013-03-01 00:00:00	0.3	0.4	1.0	NaN	1.0	18.0	64.0	7.0	NaN	2.0	NaN	0.3	280
209862	2013-03-01 00:00:00	1.0	0.3	0.4	NaN	1.0	9.0	75.0	8.0	7.0	1.0	NaN	0.5	280
209863	2013-03-01 00:00:00	NaN	NaN	NaN	0.13	2.0	11.0	73.0	NaN	NaN	NaN	1.26	NaN	280
209864	2013-03-01 00:00:00	NaN	0.5	NaN	NaN	9.0	34.0	51.0	NaN	NaN	3.0	NaN	NaN	280
209865	2013-03-01 00:00:00	NaN	0.3	NaN	NaN	1.0	6.0	NaN	4.0	NaN	25.0	NaN	NaN	280
209866	2013-03-01 00:00:00	1.0	NaN	0.4	NaN	9.0	36.0	NaN	9.0	9.0	4.0	NaN	1.9	280
209867	2013-03-01 00:00:00	NaN	0.3	NaN	NaN	1.0	16.0	70.0	NaN	NaN	NaN	NaN	NaN	280
209868	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	1.0	12.0	NaN	4.0	NaN	1.0	NaN	NaN	280
209869	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	20.0	NaN	4.0	5.0	NaN	NaN	NaN	280
209870	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	1.0	10.0	NaN	6.0	4.0	NaN	NaN	NaN	280
209871	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	1.0	14.0	70.0	NaN	NaN	NaN	NaN	NaN	280
209872	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	16.0	NaN	8.0	7.0	NaN	NaN	NaN	280
209873	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	5.0	9.0	68.0	NaN	NaN	NaN	NaN	NaN	280
209874	2013-03-01 00:00:00	0.5	NaN	1.0	0.14	3.0	18.0	NaN	2.0	NaN	NaN	1.49	0.2	280
209875	2013-03-01 00:00:00	NaN	0.4	NaN	NaN	8.0	39.0	52.0	NaN	NaN	NaN	NaN	NaN	280
209876	2013-03-01 00:00:00	NaN	0.4	NaN	NaN	1.0	11.0	NaN	6.0	NaN	2.0	NaN	NaN	280

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	s
209877	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	4.0	75.0	NaN	NaN	NaN	NaN	NaN	280
209878	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	11.0	52.0	NaN	NaN	NaN	NaN	NaN	280
209879	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	1.0	10.0	75.0	3.0	NaN	NaN	NaN	NaN	280

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

Out[5]:

	BEN	CO	EBE	NMHC	NO	NO_2	
count	50462.000000	87018.000000	50463.000000	25935.000000	209108.000000	209108.000000	1
mean	0.713075	0.328752	0.811775	0.192553	20.171921	34.710398	
std	0.841996	0.226891	0.591691	0.078111	44.385619	27.843018	
min	0.100000	0.100000	0.100000	0.040000	1.000000	1.000000	
25%	0.200000	0.200000	0.400000	0.130000	2.000000	14.000000	
50%	0.400000	0.300000	0.800000	0.180000	5.000000	27.000000	
75%	0.800000	0.400000	1.000000	0.240000	17.000000	48.000000	
max	12.100000	10.400000	11.800000	0.810000	1081.000000	388.000000	

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

Out[6]:
```
(209880, 14)
```

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

Out[7]:
```
2938320
```

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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
209875	False	True	False	True	True	False	False	False	True	True	True	True	True
209876	False	True	False	True	True	False	False	True	False	True	False	True	True
209877	False	True	True	True	True	False	False	False	True	True	True	True	True
209878	False	True	True	True	True	False	False	False	True	True	True	True	True
209879	False	True	True	True	True	False	False	False	False	True	True	True	True

209880 rows × 14 columns

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

Out[9]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	st
17286	2013-08-01 01:00:00	0.4	0.2	0.8	0.28	1.0	24.0	79.0	35.0	8.0	3.0	1.49	1.3	2807
17310	2013-08-01 02:00:00	0.5	0.2	0.9	0.28	1.0	16.0	93.0	60.0	18.0	3.0	1.61	4.0	2807
17334	2013-08-01 03:00:00	0.5	0.2	1.1	0.29	1.0	14.0	90.0	38.0	12.0	3.0	1.71	2.8	2807
17358	2013-08-01 04:00:00	0.6	0.2	1.2	0.26	1.0	12.0	84.0	30.0	8.0	3.0	1.44	2.8	2807
17382	2013-08-01 05:00:00	0.3	0.2	0.8	0.25	1.0	15.0	72.0	25.0	7.0	3.0	1.40	1.7	2807
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
209622	2013-02-28 14:00:00	1.1	0.3	0.3	0.27	3.0	17.0	64.0	5.0	5.0	2.0	1.41	0.9	2807
209646	2013-02-28 15:00:00	1.3	0.4	0.3	0.27	2.0	16.0	66.0	6.0	5.0	1.0	1.40	0.9	2807
209670	2013-02-28 16:00:00	1.1	0.3	0.3	0.27	1.0	17.0	65.0	5.0	4.0	1.0	1.40	0.7	2807
209694	2013-02-28 17:00:00	1.0	0.3	0.4	0.27	1.0	18.0	64.0	5.0	5.0	1.0	1.39	0.7	2807
209718	2013-02-28 18:00:00	1.0	0.3	0.4	0.27	1.0	22.0	62.0	6.0	6.0	1.0	1.39	0.7	2807

7315 rows × 14 columns

In [10]:
```
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]:
```
sd=data[['BEN','CO', 'EBE', 'NMHC', 'NO_2']]
```

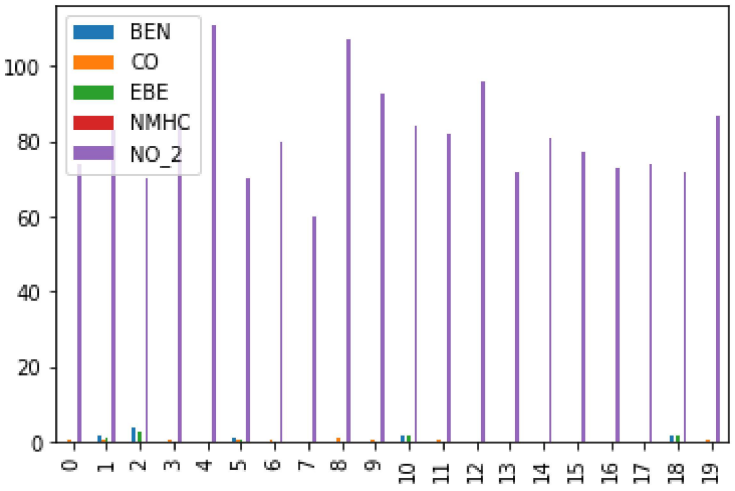
In [12]:
```
dd=sd.head(20)
dd
```

Out[12]:

	BEN	CO	EBE	NMHC	NO_2
0	NaN	0.6	NaN	NaN	74.0
1	1.5	0.5	1.3	NaN	83.0
2	3.9	NaN	2.8	NaN	70.0
3	NaN	0.5	NaN	NaN	87.0
4	NaN	NaN	NaN	NaN	111.0
5	1.0	0.6	0.8	NaN	70.0
6	NaN	0.4	NaN	0.29	80.0
7	NaN	NaN	NaN	0.23	60.0
8	NaN	1.0	NaN	NaN	107.0
9	NaN	0.6	NaN	NaN	93.0
10	1.4	NaN	1.4	NaN	84.0
11	NaN	0.6	NaN	NaN	82.0
12	NaN	NaN	NaN	NaN	96.0
13	NaN	NaN	NaN	NaN	72.0
14	NaN	NaN	NaN	NaN	81.0
15	NaN	NaN	NaN	NaN	77.0
16	NaN	NaN	NaN	NaN	73.0
17	NaN	NaN	NaN	NaN	74.0
18	1.6	NaN	1.4	0.22	72.0
19	NaN	0.8	NaN	NaN	87.0

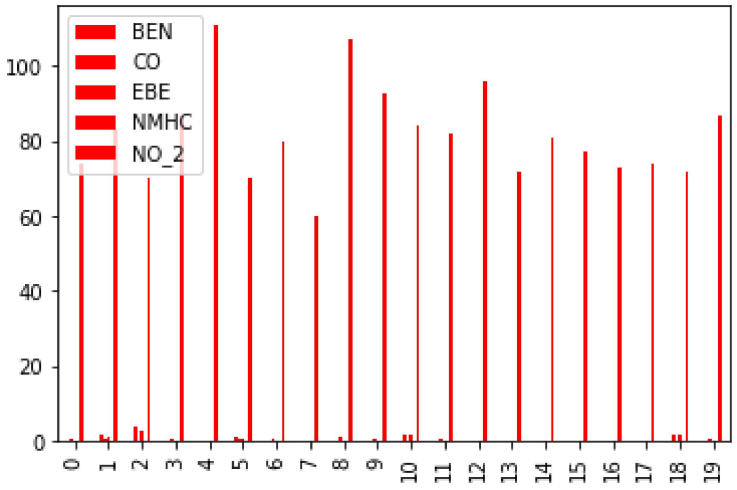
In [13]:
```
dd.plot.bar()
```

Out[13]: <AxesSubplot:>



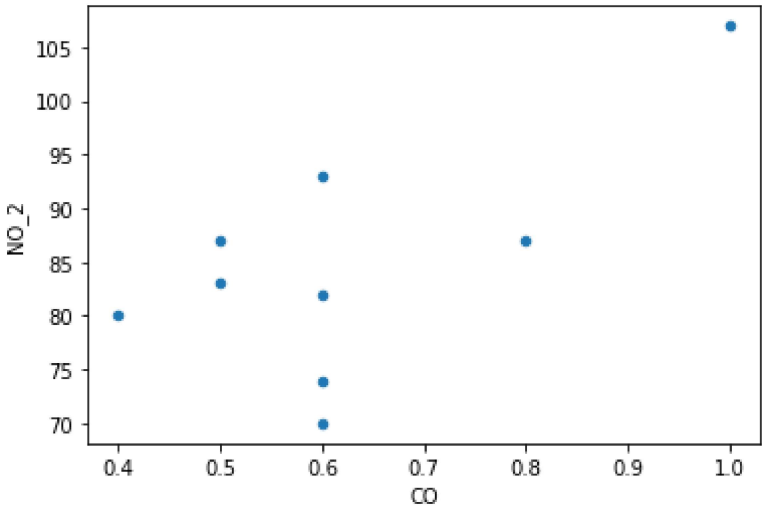
In [14]:
```
dd.plot.bar(color='r')
```

Out[14]: <AxesSubplot:>



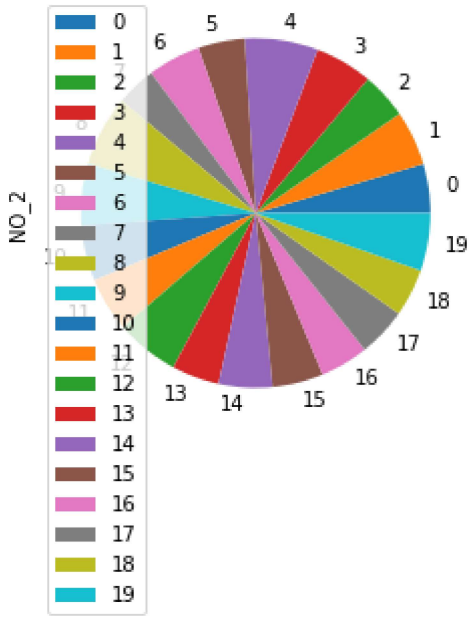
In [15]:
```
dd.plot.scatter(x='CO',y='NO_2')
```

Out[15]: <AxesSubplot:xlabel='CO', ylabel='NO_2'>



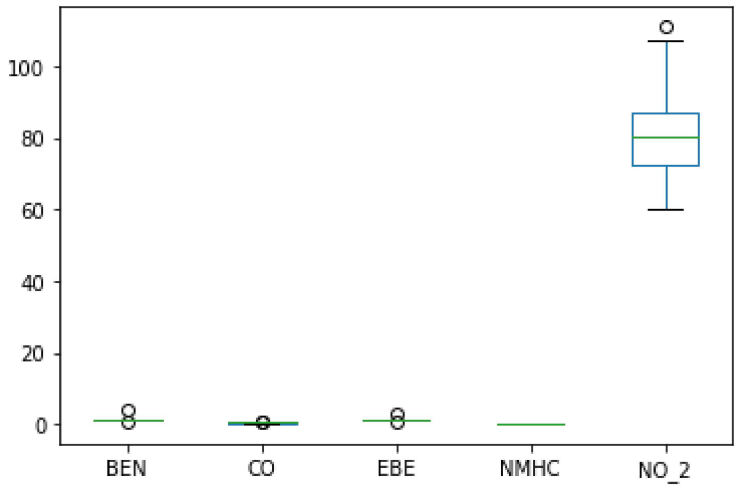
In [16]:
```
dd.plot.pie(y='NO_2')
```

Out[16]: <AxesSubplot:ylabel='NO_2'>



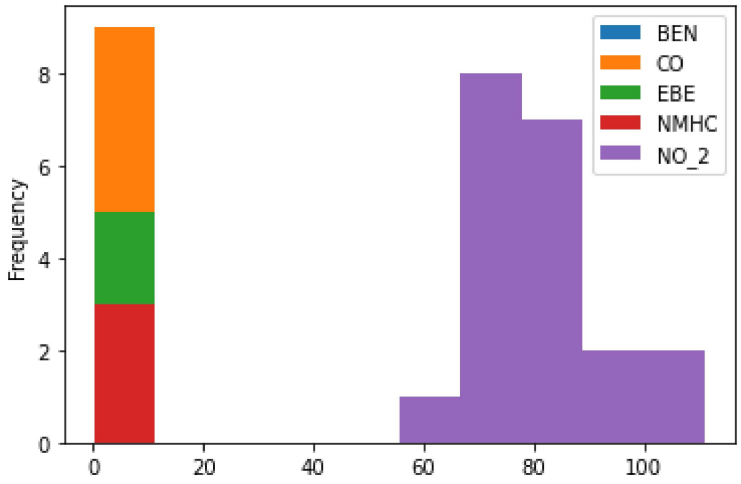
In [17]:
```
dd.plot.box()
```

Out[17]: <AxesSubplot:>



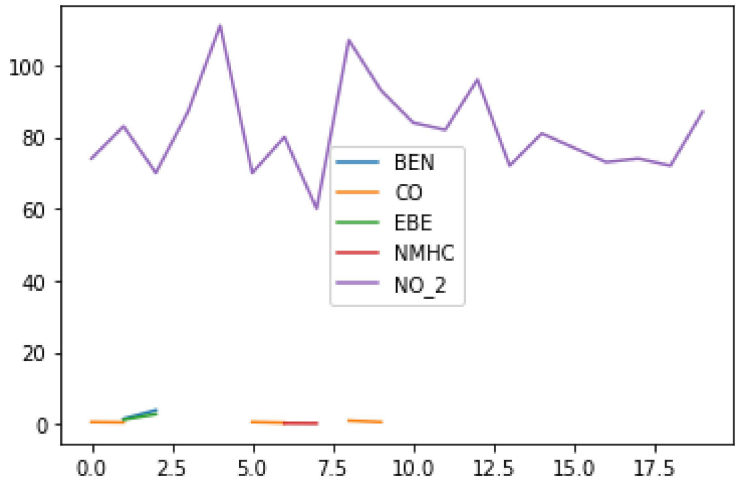
In [18]:
```
dd.plot.hist()
```

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



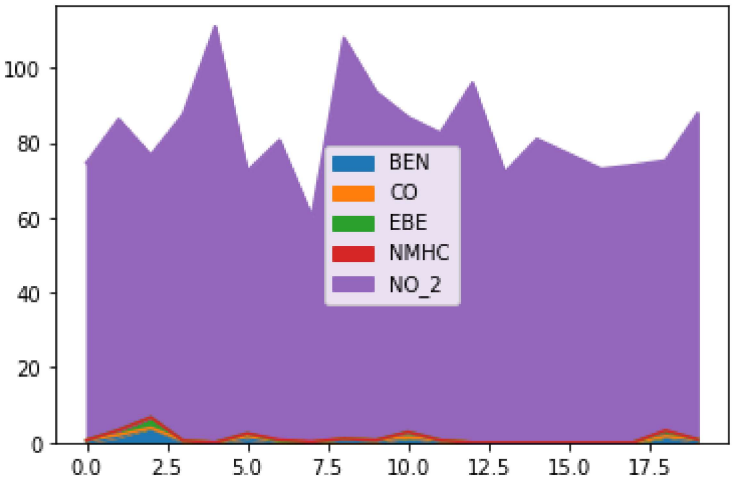
In [19]:
```
dd.plot.line()
```

Out[19]: <AxesSubplot:>



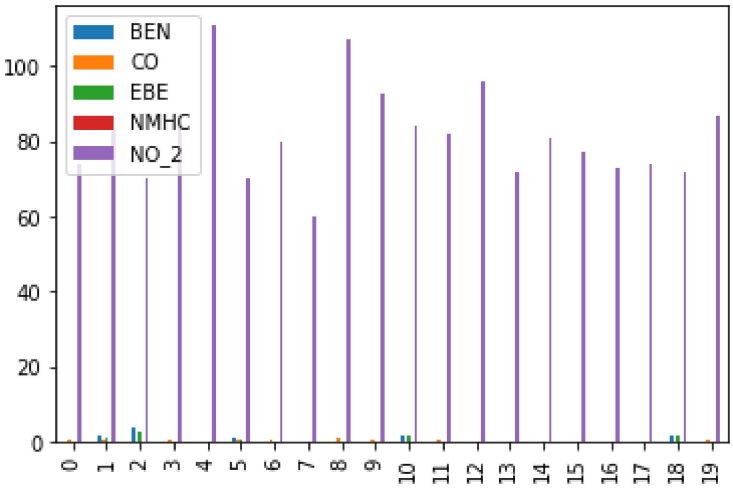
In [20]:
```
dd.plot.area()
```

Out[20]: <AxesSubplot:>



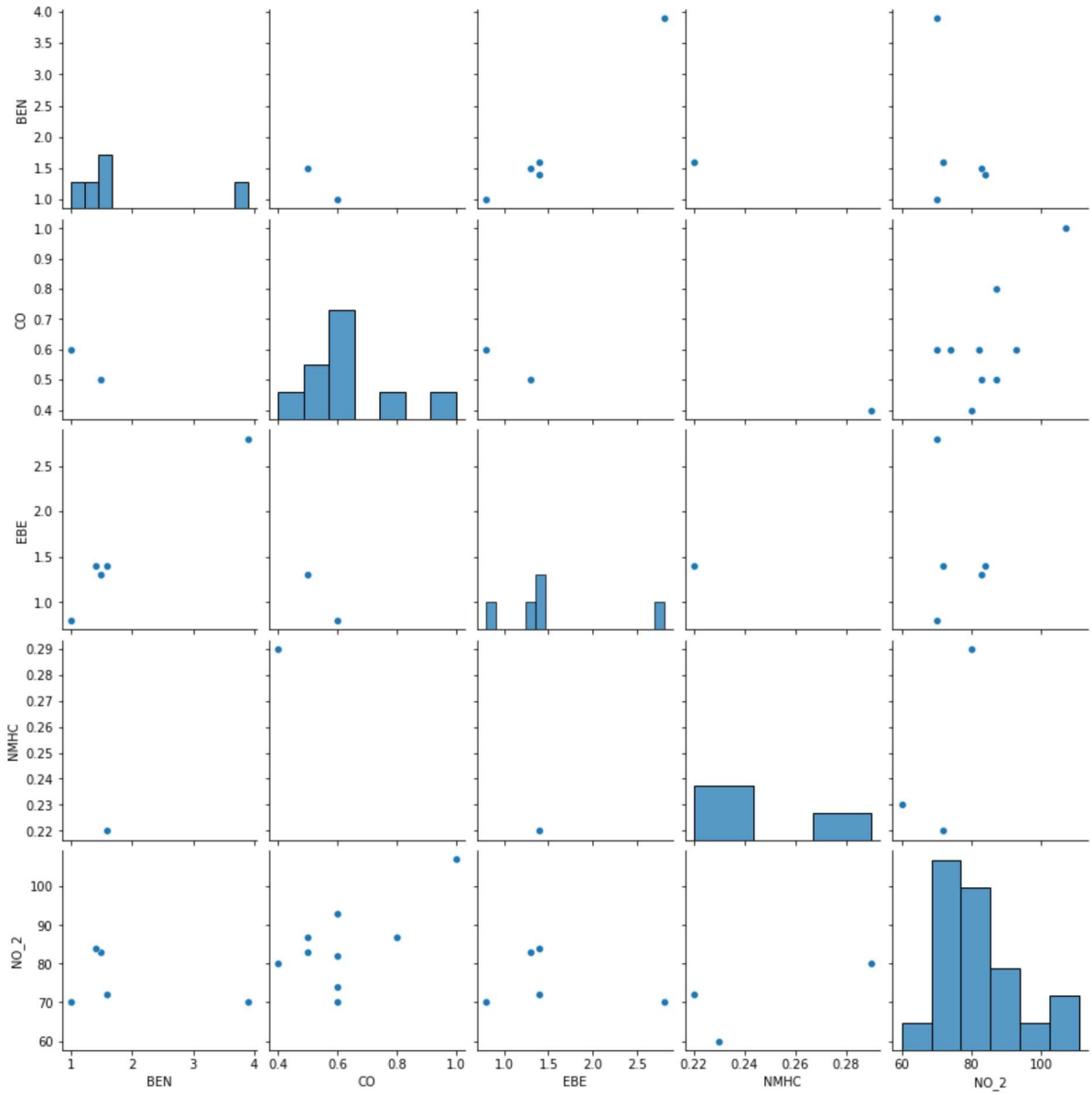
In [21]:
```
dd.plot.bar()
```

Out[21]: <AxesSubplot:>



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

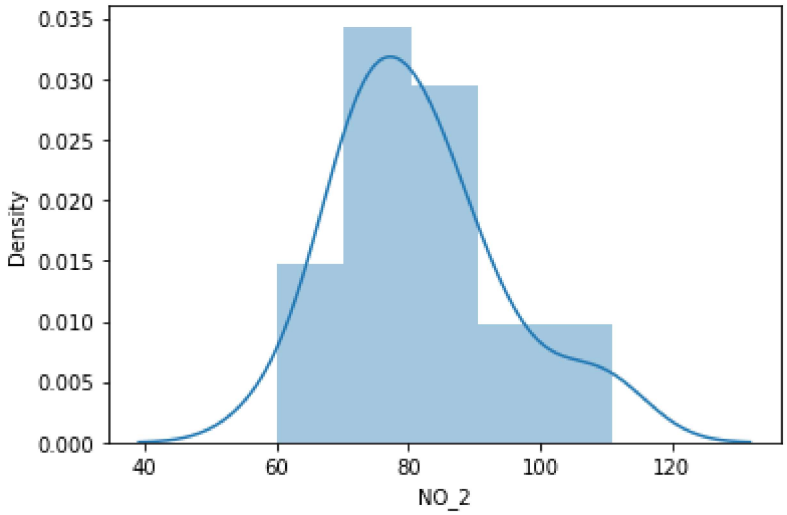
Out[22]: <seaborn.axisgrid.PairGrid at 0x1d32f983190>



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

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

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



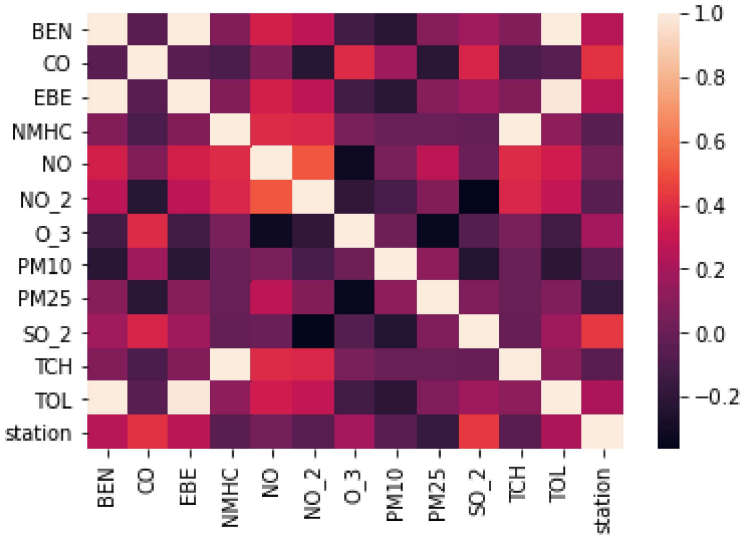
In [24]:
```
ds=data.fillna(20)
```

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_)
```

```
28078950.831174836
```

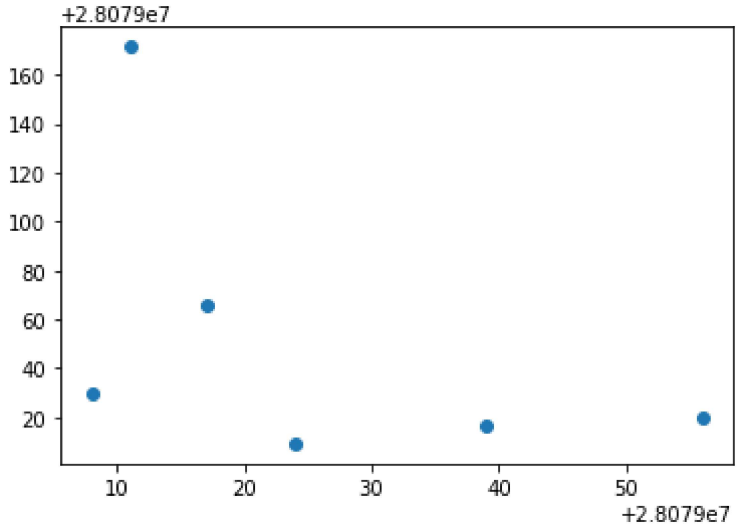
In [32]:
```
coeff= pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[32]:

	Co-efficient
BEN	127.453823
CO	1.690588
EBE	-126.754534
NMHC	0.317007
NO_2	0.546350

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

Out[33]:
```
<matplotlib.collections.PathCollection at 0x1d332a6c070>
```



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

```
-16.943459842274965
```

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

Out[35]:
```
-16.943459842274965
```

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

Out[36]:
```
0.7632894632603324
```

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]:
```
-1.7259686522788047
```

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

Out[40]:
```
0.6654810806033797
```

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]:
```
-1.3303074679288511
```

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

Out[43]:
```
0.6494607183239762
```

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.          1.4141002  -0.08879058  0.00622733  0.41983051]
```

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

```
28078986.138676934
```

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

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

```
-1.7000130019693187
```

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)
```

```
[28079017]
```

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]:
```
5.741483976283094e-05
```

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]:
```
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 [66]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[66]:
```
RandomForestClassifier()
```

In [67]:
```
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 [68]:
```
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[68]:
```
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')
```

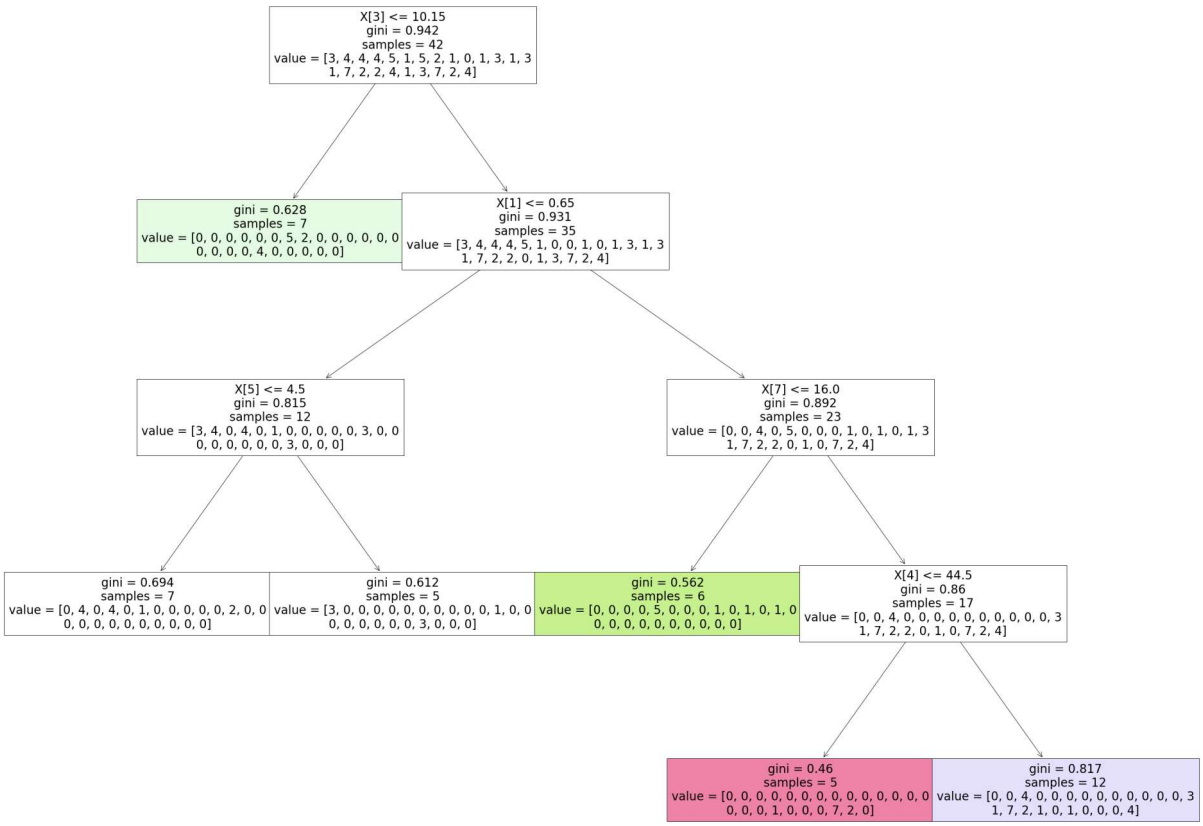
In [69]:
```
grid_search.best_score_
```

Out[69]:
```
0.5285714285714285
```

In [70]:
```
rfc_best=grid_search.best_estimator_
```

In [71]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(50,40))
plot_tree(rfc_best.estimators_[5],filled=True)
```

Out[71]:
```
[Text(930.0, 1956.96, 'X[3] <= 10.15\ngini = 0.942\nsamples = 42\nvalue = [3,
4, 4, 4, 5, 1, 5, 2, 1, 0, 1, 3, 1, 3\n1, 7, 2, 2, 4, 1, 3, 7, 2, 4]'),
 Text(620.0, 1522.0800000000002, 'gini = 0.628\nsamples = 7\nvalue = [0, 0,
0, 0, 0, 0, 5, 2, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 4, 0, 0, 0, 0, 0]'),
 Text(1240.0, 1522.0800000000002, 'X[1] <= 0.65\ngini = 0.931\nsamples = 35\n
value = [3, 4, 4, 4, 5, 1, 0, 0, 1, 0, 1, 3, 1, 3\n1, 7, 2, 2, 0, 1, 3, 7, 2,
4]'),
 Text(620.0, 1087.2, 'X[5] <= 4.5\ngini = 0.815\nsamples = 12\nvalue = [3, 4,
0, 4, 0, 1, 0, 0, 0, 0, 0, 3, 0, 0\n0, 0, 0, 0, 0, 0, 3, 0, 0, 0]'),
 Text(310.0, 652.3200000000002, 'gini = 0.694\nsamples = 7\nvalue = [0, 4, 0,
4, 0, 1, 0, 0, 0, 0, 0, 2, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(930.0, 652.3200000000002, 'gini = 0.612\nsamples = 5\nvalue = [3, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0\n0, 0, 0, 0, 0, 0, 3, 0, 0, 0]'),
 Text(1860.0, 1087.2, 'X[7] <= 16.0\ngini = 0.892\nsamples = 23\nvalue = [0,
0, 4, 0, 5, 0, 0, 0, 1, 0, 1, 0, 1, 3\n1, 7, 2, 2, 0, 1, 0, 7, 2, 4]'),
 Text(1550.0, 652.3200000000002, 'gini = 0.562\nsamples = 6\nvalue = [0, 0,
0, 0, 5, 0, 0, 0, 1, 0, 1, 0, 1, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(2170.0, 652.3200000000002, 'X[4] <= 44.5\ngini = 0.86\nsamples = 17\nva
lue = [0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3\n1, 7, 2, 2, 0, 1, 0, 7, 2,
4]'),
 Text(1860.0, 217.44000000000005, 'gini = 0.46\nsamples = 5\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 1, 0, 0, 0, 7, 2, 0]'),
 Text(2480.0, 217.44000000000005, 'gini = 0.817\nsamples = 12\nvalue = [0, 0,
4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3\n1, 7, 2, 1, 0, 1, 0, 0, 0, 4]')]
```



Conclusion : LinearRegression()
28078950.831174836 HIGH RANGE

In [ ]: