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_2017.csv")
data
```

Out[2]:

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2	TCI
0	2017-06-01 01:00:00	NaN	NaN	0.3	NaN	NaN	4.0	38.0	NaN	NaN	NaN	NaN	5.0	Nal
1	2017-06-01 01:00:00	0.6	NaN	0.3	0.4	0.08	3.0	39.0	NaN	71.0	22.0	9.0	7.0	1.4
2	2017-06-01 01:00:00	0.2	NaN	NaN	0.1	NaN	1.0	14.0	NaN	NaN	NaN	NaN	NaN	Nal
3	2017-06-01 01:00:00	NaN	NaN	0.2	NaN	NaN	1.0	9.0	NaN	91.0	NaN	NaN	NaN	Nal
4	2017-06-01 01:00:00	NaN	NaN	NaN	NaN	NaN	1.0	19.0	NaN	69.0	NaN	NaN	2.0	Nal
...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
210115	2017-08-01 00:00:00	NaN	NaN	0.2	NaN	NaN	1.0	27.0	NaN	65.0	NaN	NaN	NaN	Nal
210116	2017-08-01 00:00:00	NaN	NaN	0.2	NaN	NaN	1.0	14.0	NaN	NaN	73.0	NaN	7.0	Nal
210117	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	4.0	NaN	83.0	NaN	NaN	NaN	Nal
210118	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	11.0	NaN	78.0	NaN	NaN	NaN	Nal
210119	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	14.0	NaN	77.0	60.0	NaN	NaN	Nal

210120 rows × 16 columns

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

Out[3]:

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2	TCH	TO
0	2017-06-01 01:00:00	NaN	NaN	0.3	NaN	NaN	4.0	38.0	NaN	NaN	NaN	NaN	5.0	NaN	Na
1	2017-06-01 01:00:00	0.6	NaN	0.3	0.4	0.08	3.0	39.0	NaN	71.0	22.0	9.0	7.0	1.40	2
2	2017-06-01 01:00:00	0.2	NaN	NaN	0.1	NaN	1.0	14.0	NaN	NaN	NaN	NaN	NaN	NaN	0
3	2017-06-01 01:00:00	NaN	NaN	0.2	NaN	NaN	1.0	9.0	NaN	91.0	NaN	NaN	NaN	NaN	Na
4	2017-06-01 01:00:00	NaN	NaN	NaN	NaN	NaN	1.0	19.0	NaN	69.0	NaN	NaN	2.0	NaN	Na
5	2017-06-01 01:00:00	0.1	NaN	0.3	0.2	NaN	1.0	26.0	NaN	70.0	26.0	NaN	1.0	NaN	0
6	2017-06-01 01:00:00	0.3	NaN	0.2	0.1	0.17	1.0	19.0	NaN	79.0	23.0	9.0	3.0	0.86	1
7	2017-06-01 01:00:00	NaN	NaN	NaN	NaN	NaN	1.0	9.0	NaN	87.0	NaN	NaN	NaN	NaN	Na
8	2017-06-01 01:00:00	NaN	NaN	0.3	NaN	NaN	3.0	30.0	NaN	70.0	NaN	NaN	NaN	NaN	Na
9	2017-06-01 01:00:00	NaN	NaN	0.1	NaN	NaN	1.0	15.0	NaN	NaN	22.0	NaN	10.0	NaN	Na

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

Out[4]:

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2	TC
210100	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	39.0	NaN	55.0	NaN	NaN	1.0	Na
210101	2017-08-01 00:00:00	0.3	NaN	0.2	0.2	NaN	1.0	23.0	NaN	61.0	33.0	NaN	1.0	Na
210102	2017-08-01 00:00:00	0.1	NaN	0.1	0.1	0.08	1.0	10.0	NaN	77.0	28.0	10.0	3.0	1.1
210103	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	23.0	NaN	69.0	NaN	NaN	NaN	Na
210104	2017-08-01 00:00:00	NaN	NaN	0.2	NaN	NaN	5.0	39.0	NaN	37.0	NaN	NaN	9.0	Na
210105	2017-08-01 00:00:00	NaN	NaN	0.2	NaN	NaN	2.0	25.0	NaN	NaN	27.0	NaN	5.0	Na
210106	2017-08-01 00:00:00	0.3	NaN	NaN	0.4	NaN	3.0	20.0	NaN	NaN	25.0	10.0	2.0	Na
210107	2017-08-01 00:00:00	NaN	NaN	0.2	NaN	NaN	2.0	15.0	NaN	69.0	NaN	NaN	NaN	Na
210108	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	2.0	27.0	NaN	NaN	34.0	NaN	8.0	Na
210109	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	24.0	NaN	NaN	24.0	15.0	NaN	Na
210110	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	2.0	19.0	NaN	NaN	23.0	11.0	NaN	Na
210111	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	15.0	NaN	57.0	NaN	NaN	NaN	Na
210112	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	15.0	29.0	NaN	NaN	23.0	10.0	NaN	Na
210113	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	2.0	24.0	NaN	71.0	NaN	NaN	NaN	Na
210114	2017-08-01 00:00:00	0.3	NaN	NaN	0.4	0.09	1.0	18.0	NaN	NaN	23.0	NaN	NaN	1.3
210115	2017-08-01 00:00:00	NaN	NaN	0.2	NaN	NaN	1.0	27.0	NaN	65.0	NaN	NaN	NaN	Na
210116	2017-08-01 00:00:00	NaN	NaN	0.2	NaN	NaN	1.0	14.0	NaN	NaN	73.0	NaN	7.0	Na

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2	TC
210117	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	4.0	NaN	83.0	NaN	NaN	NaN	Na
210118	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	11.0	NaN	78.0	NaN	NaN	NaN	Na
210119	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	14.0	NaN	77.0	60.0	NaN	NaN	Na

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

Out[5]:

	BEN	CH4	CO	EBE	NMHC	NO	
count	50201.000000	6410.000000	87001.000000	49973.000000	25472.000000	209065.000000	209(
mean	0.595534	1.321218	0.363866	0.394819	0.127865	23.409189	
std	0.774482	0.203652	0.262726	0.674253	0.094632	50.362967	
min	0.100000	1.000000	0.100000	0.100000	0.000000	1.000000	
25%	0.200000	1.190000	0.200000	0.100000	0.080000	2.000000	
50%	0.400000	1.280000	0.300000	0.200000	0.110000	6.000000	
75%	0.700000	1.390000	0.400000	0.500000	0.160000	20.000000	
max	19.600000	3.630000	4.900000	38.299999	4.400000	973.000000	

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

Out[6]:
```
(210120, 16)
```

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

Out[7]:
```
3361920
```

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

Out[8]:

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2	
0	False	True	True	False	True	True	False	False	True	True	True	True	False	
1	False	False	True	False	False	False	False	False	True	False	False	False	False	F
2	False	False	True	True	False	True	False	False	True	True	True	True	True	
3	False	True	True	False	True	True	False	False	True	False	True	True	True	
4	False	True	True	True	True	True	False	False	True	False	True	True	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210115	False	True	True	False	True	True	False	False	True	False	True	True	True	
210116	False	True	True	False	True	True	False	False	True	True	False	True	False	
210117	False	True	True	True	True	True	False	False	True	False	True	True	True	
210118	False	True	True	True	True	True	False	False	True	False	True	True	True	
210119	False	True	True	True	True	True	False	False	True	False	False	True	True	

210120 rows × 16 columns

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

Out[9]:

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2	TCH
87457	2017-10-01 01:00:00	0.6	1.22	0.3	0.4	0.09	4.0	54.0	60.0	43.0	12.0	9.0	13.0	1.31
87462	2017-10-01 01:00:00	0.2	1.18	0.2	0.1	0.09	1.0	26.0	28.0	42.0	14.0	6.0	3.0	1.27
87481	2017-10-01 02:00:00	0.4	1.22	0.2	0.2	0.06	2.0	32.0	36.0	53.0	14.0	10.0	13.0	1.28
87486	2017-10-01 02:00:00	0.2	1.19	0.2	0.1	0.07	1.0	15.0	17.0	51.0	18.0	8.0	3.0	1.26
87505	2017-10-01 03:00:00	0.3	1.23	0.2	0.2	0.06	2.0	27.0	29.0	57.0	15.0	10.0	13.0	1.29
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
158238	2017-12-31 22:00:00	0.3	1.11	0.2	0.1	0.03	1.0	8.0	9.0	73.0	3.0	1.0	3.0	1.14
158257	2017-12-31 23:00:00	0.6	1.38	0.3	0.1	0.03	6.0	42.0	51.0	47.0	7.0	4.0	3.0	1.41
158262	2017-12-31 23:00:00	0.3	1.11	0.2	0.1	0.03	1.0	6.0	8.0	72.0	6.0	3.0	3.0	1.14
158281	2018-01-01 00:00:00	0.5	1.38	0.2	0.1	0.02	2.0	20.0	23.0	69.0	4.0	2.0	3.0	1.39
158286	2018-01-01 00:00:00	0.3	1.11	0.2	0.1	0.03	1.0	1.0	3.0	83.0	8.0	5.0	3.0	1.14

4127 rows × 16 columns

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

Out[10]:
```
Index(['date', 'BEN', 'CH4', 'CO', 'EBE', 'NMHC', 'NO', 'NO_2', 'NOx', 'O_3',
       'PM10', 'PM25', 'SO_2', 'TCH', 'TOL', 'station'],
      dtype='object')
```

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

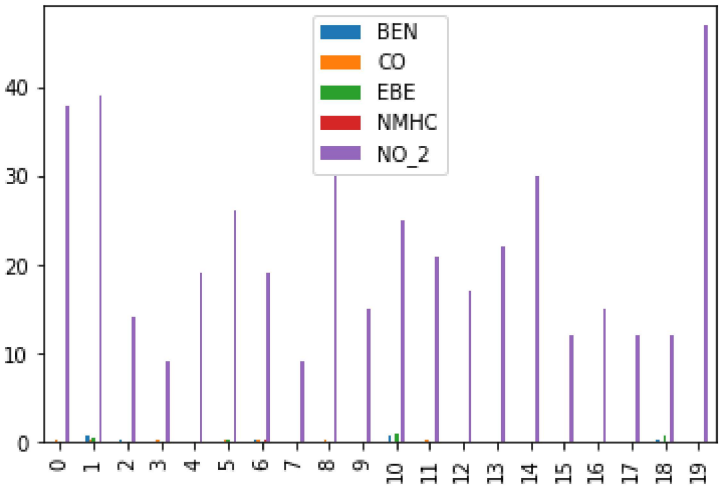
In [12]:
```
dd=sd.head(20)
dd
```

Out[12]:

	BEN	CO	EBE	NMHC	NO_2
0	NaN	0.3	NaN	NaN	38.0
1	0.6	0.3	0.4	0.08	39.0
2	0.2	NaN	0.1	NaN	14.0
3	NaN	0.2	NaN	NaN	9.0
4	NaN	NaN	NaN	NaN	19.0
5	0.1	0.3	0.2	NaN	26.0
6	0.3	0.2	0.1	0.17	19.0
7	NaN	NaN	NaN	NaN	9.0
8	NaN	0.3	NaN	NaN	30.0
9	NaN	0.1	NaN	NaN	15.0
10	0.7	NaN	1.0	NaN	25.0
11	NaN	0.2	NaN	NaN	21.0
12	NaN	NaN	NaN	NaN	17.0
13	NaN	NaN	NaN	NaN	22.0
14	NaN	NaN	NaN	NaN	30.0
15	NaN	NaN	NaN	NaN	12.0
16	NaN	NaN	NaN	NaN	15.0
17	NaN	NaN	NaN	NaN	12.0
18	0.2	NaN	0.6	0.08	12.0
19	NaN	0.1	NaN	NaN	47.0

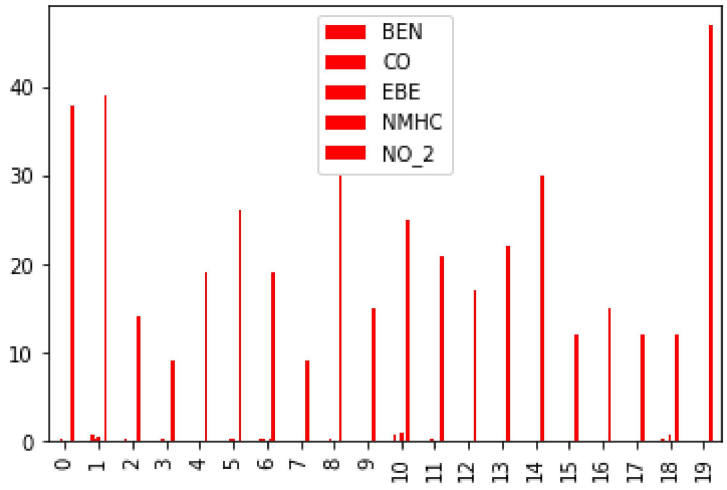
In [13]:
```
dd.plot.bar()
```

Out[13]: <AxesSubplot:>



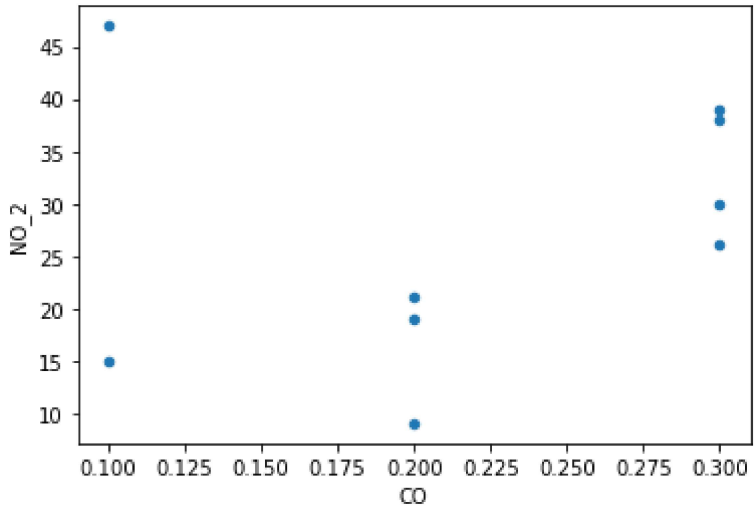
In [14]:
```
dd.plot.bar(color='r')
```

Out[14]: <AxesSubplot:>



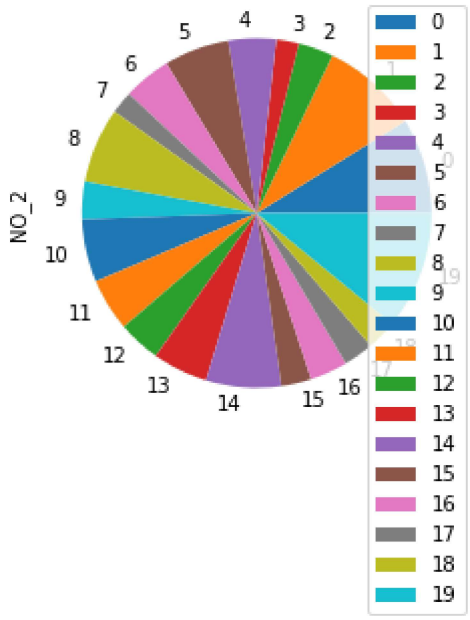
In [15]:
```
dd.plot.scatter(x='CO',y='NO_2')
```

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



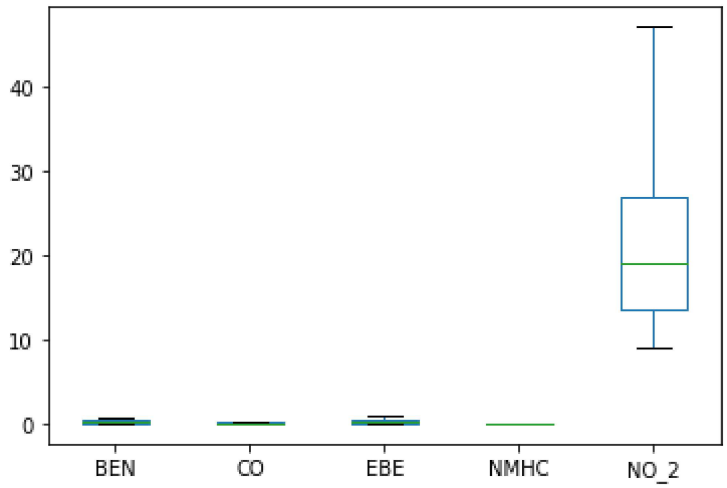
In [16]:
```
dd.plot.pie(y='NO_2')
```

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



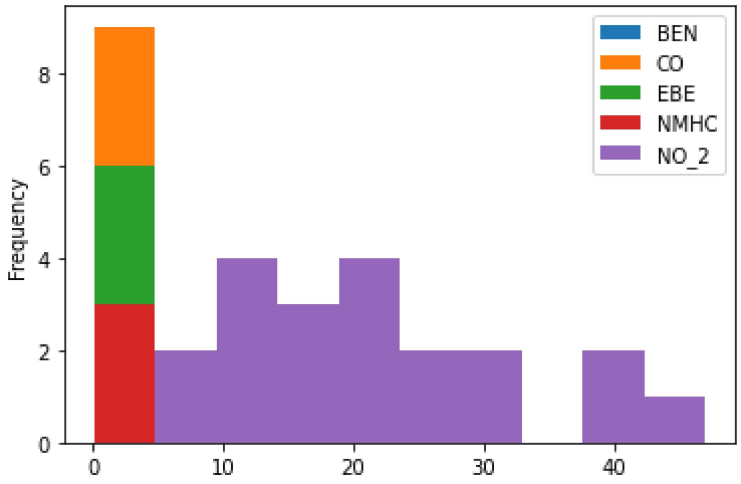
In [17]:
```
dd.plot.box()
```

Out[17]: <AxesSubplot:>



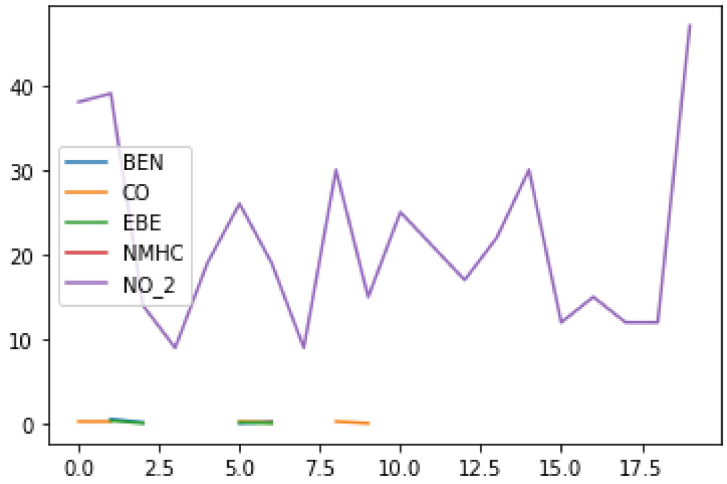
In [18]:
```
dd.plot.hist()
```

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



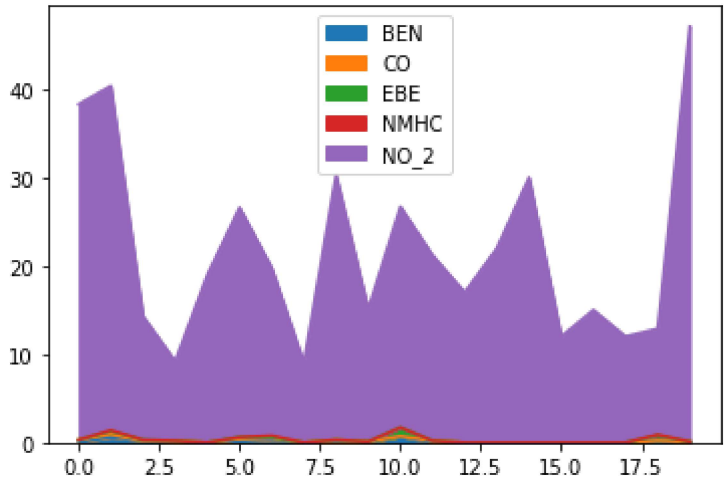
In [19]:
```
dd.plot.line()
```

Out[19]: <AxesSubplot:>



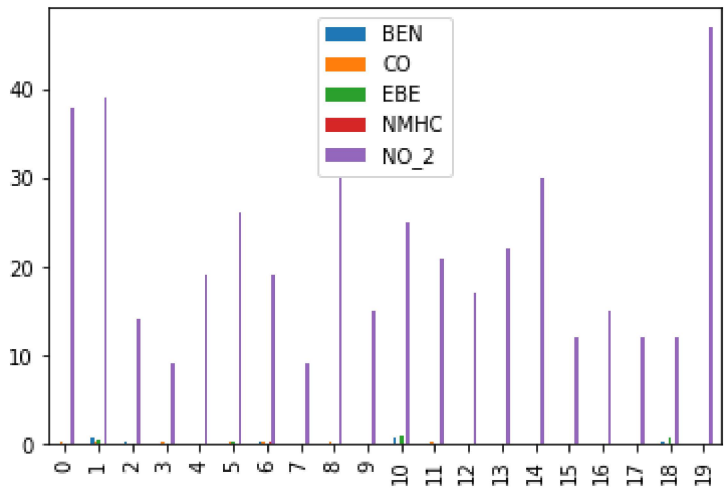
In [20]:
```
dd.plot.area()
```

Out[20]: <AxesSubplot:>



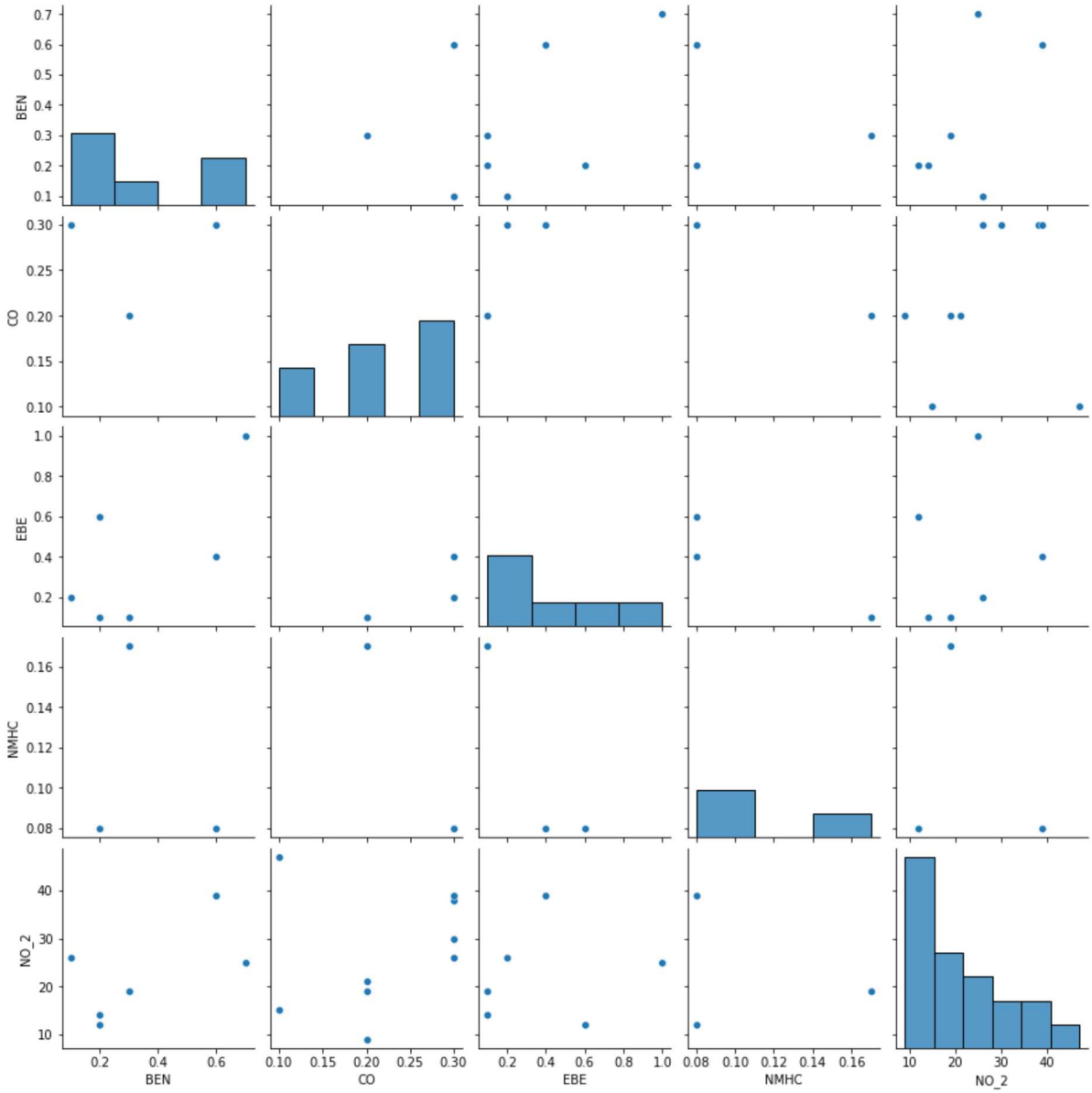
In [21]:
```
dd.plot.bar()
```

Out[21]: <AxesSubplot:>



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

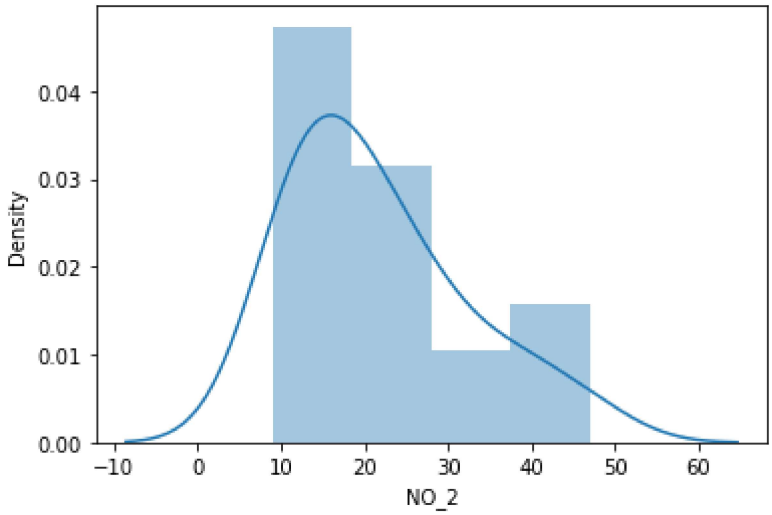
Out[22]:
```
<seaborn.axisgrid.PairGrid at 0x268351f2880>
```



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'>
```



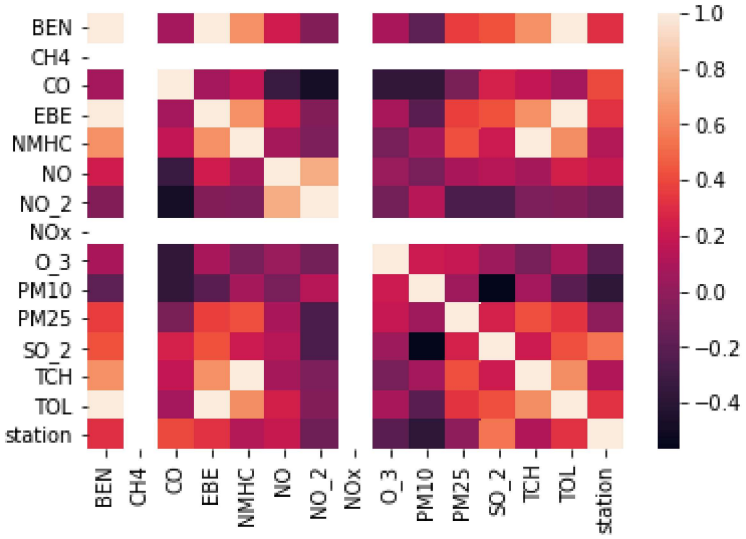
In [24]:
```
ds=data.fillna(20)
```

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_)
```

```
28079017.74650614
```

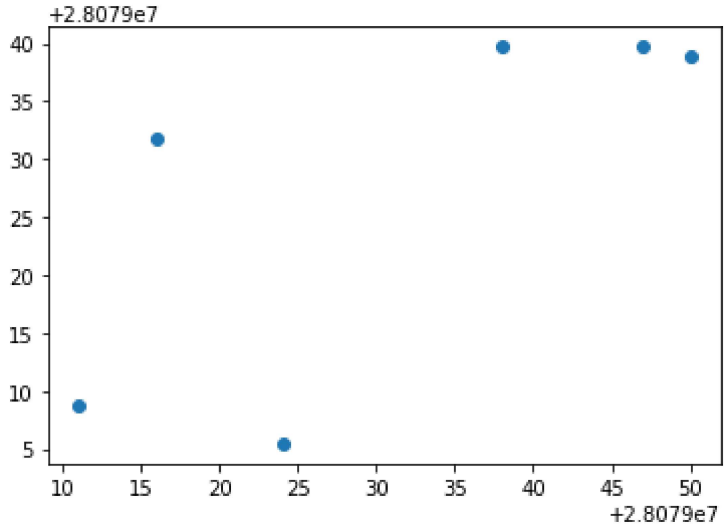
In [32]:
```
coeff= pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[32]:

	Co-efficient
BEN	-71.896181
CO	0.328882
EBE	73.049409
NMHC	-0.500365
NO_2	0.105457

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

Out[33]:
```
<matplotlib.collections.PathCollection at 0x26839df90a0>
```



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

```
0.42030487490222956
```

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

Out[35]:
```
0.42030487490222956
```

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

Out[36]:
```
0.3858736284761375
```

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]:
```
0.37447813710370226
```

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

Out[40]:
```
0.2024649914896749
```

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]:
```
0.23638307519831647
```

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

Out[43]:
```
0.18157828705094115
```

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.          0.47829967  0.62436811 -0.37232379 -0.12919979]
```

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

```
28079029.41629657
```

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

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

```
0.3704720814228958
```

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]:
```
0.05997604883753372
```

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)
```

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')
```

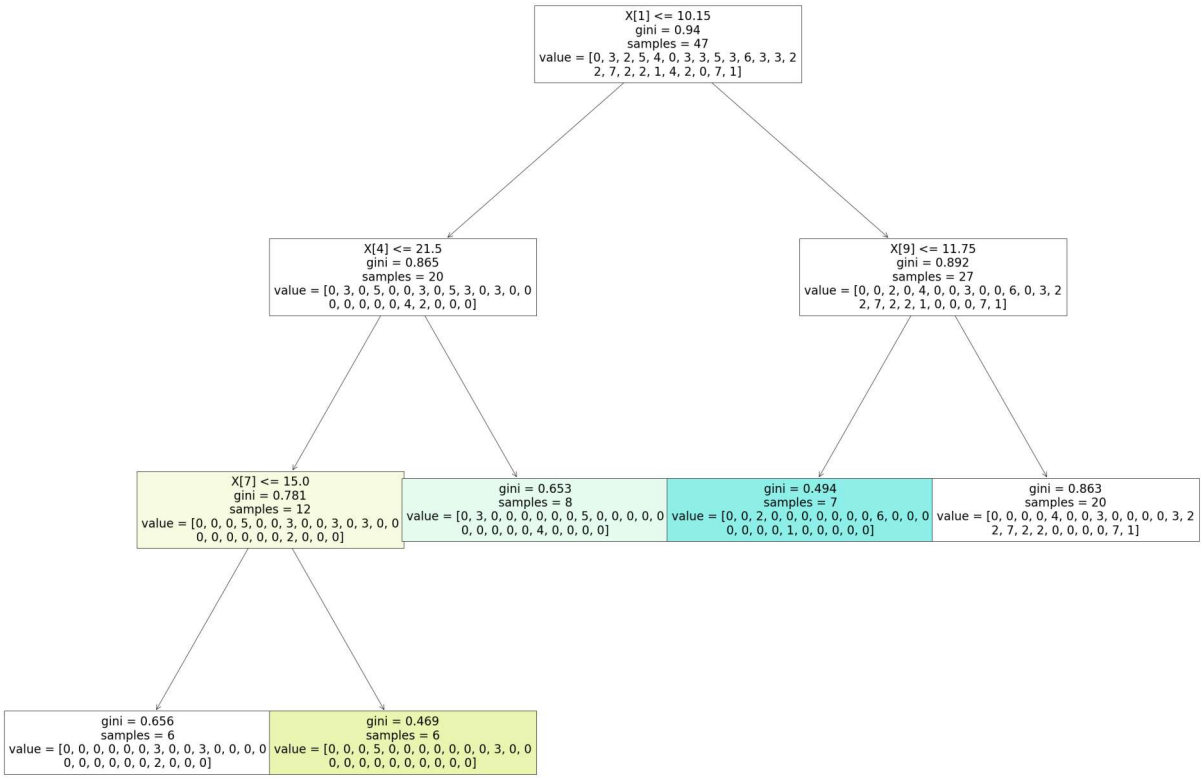
In [69]:
```
grid_search.best_score_
```

Out[69]:
```
0.45714285714285713
```

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(1550.0, 1902.6000000000001, 'X[1] <= 10.15\ngini = 0.94\nsamples = 47\n
value = [0, 3, 2, 5, 4, 0, 3, 3, 5, 3, 6, 3, 3, 2\n2, 7, 2, 2, 1, 4, 2, 0, 7,
1]'),
 Text(930.0, 1359.0, 'X[4] <= 21.5\ngini = 0.865\nsamples = 20\nvalue = [0,
3, 0, 5, 0, 0, 3, 0, 5, 3, 0, 3, 0, 0\n0, 0, 0, 0, 0, 4, 2, 0, 0, 0]'),
 Text(620.0, 815.4000000000001, 'X[7] <= 15.0\ngini = 0.781\nsamples = 12\nva
lue = [0, 0, 0, 5, 0, 0, 3, 0, 0, 3, 0, 3, 0, 0\n0, 0, 0, 0, 0, 0, 2, 0, 0,
0]'),
 Text(310.0, 271.79999999999995, 'gini = 0.656\nsamples = 6\nvalue = [0, 0,
0, 0, 0, 0, 3, 0, 0, 3, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 2, 0, 0, 0]'),
 Text(930.0, 271.79999999999995, 'gini = 0.469\nsamples = 6\nvalue = [0, 0,
0, 5, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1240.0, 815.4000000000001, 'gini = 0.653\nsamples = 8\nvalue = [0, 3,
0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 4, 0, 0, 0, 0]'),
 Text(2170.0, 1359.0, 'X[9] <= 11.75\ngini = 0.892\nsamples = 27\nvalue = [0,
0, 2, 0, 4, 0, 0, 3, 0, 0, 6, 0, 3, 2\n2, 7, 2, 2, 1, 0, 0, 0, 7, 1]'),
 Text(1860.0, 815.4000000000001, 'gini = 0.494\nsamples = 7\nvalue = [0, 0,
2, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0\n0, 0, 0, 0, 1, 0, 0, 0, 0, 0]'),
 Text(2480.0, 815.4000000000001, 'gini = 0.863\nsamples = 20\nvalue = [0, 0,
0, 0, 4, 0, 0, 3, 0, 0, 0, 0, 3, 2\n2, 7, 2, 2, 0, 0, 0, 0, 7, 1]')]
```



Conclusion : LogisticRegression() predict 0.05997604883753372 HIGH RANGE

In [ ]: