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_2011.csv")
data
```

Out[2]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
0	2011-11-01 01:00:00	NaN	1.0	NaN	NaN	154.0	84.0	NaN	NaN	NaN	6.0	NaN	NaN	2
1	2011-11-01 01:00:00	2.5	0.4	3.5	0.26	68.0	92.0	3.0	40.0	24.0	9.0	1.54	8.7	2
2	2011-11-01 01:00:00	2.9	NaN	3.8	NaN	96.0	99.0	NaN	NaN	NaN	NaN	NaN	7.2	2
3	2011-11-01 01:00:00	NaN	0.6	NaN	NaN	60.0	83.0	2.0	NaN	NaN	NaN	NaN	NaN	2
4	2011-11-01 01:00:00	NaN	NaN	NaN	NaN	44.0	62.0	3.0	NaN	NaN	3.0	NaN	NaN	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
209923	2011-09-01 00:00:00	NaN	0.2	NaN	NaN	5.0	19.0	44.0	NaN	NaN	NaN	NaN	NaN	2
209924	2011-09-01 00:00:00	NaN	0.1	NaN	NaN	6.0	29.0	NaN	11.0	NaN	7.0	NaN	NaN	2
209925	2011-09-01 00:00:00	NaN	NaN	NaN	0.23	1.0	21.0	28.0	NaN	NaN	NaN	1.44	NaN	2
209926	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	3.0	15.0	48.0	NaN	NaN	NaN	NaN	NaN	2
209927	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	4.0	33.0	38.0	13.0	NaN	NaN	NaN	NaN	2

209928 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	2011-11-01 01:00:00	NaN	1.0	NaN	NaN	154.0	84.0	NaN	NaN	NaN	6.0	NaN	NaN	280790
1	2011-11-01 01:00:00	2.5	0.4	3.5	0.26	68.0	92.0	3.0	40.0	24.0	9.0	1.54	8.7	280790
2	2011-11-01 01:00:00	2.9	NaN	3.8	NaN	96.0	99.0	NaN	NaN	NaN	NaN	NaN	7.2	280790
3	2011-11-01 01:00:00	NaN	0.6	NaN	NaN	60.0	83.0	2.0	NaN	NaN	NaN	NaN	NaN	280790
4	2011-11-01 01:00:00	NaN	NaN	NaN	NaN	44.0	62.0	3.0	NaN	NaN	3.0	NaN	NaN	280790
5	2011-11-01 01:00:00	0.5	0.8	0.3	NaN	102.0	75.0	2.0	35.0	NaN	5.0	NaN	4.3	280790
6	2011-11-01 01:00:00	0.7	0.3	1.1	0.16	17.0	66.0	7.0	22.0	16.0	2.0	1.36	1.7	280790
7	2011-11-01 01:00:00	NaN	NaN	NaN	0.36	83.0	78.0	6.0	NaN	NaN	NaN	1.80	NaN	280790
8	2011-11-01 01:00:00	NaN	0.7	NaN	NaN	80.0	91.0	5.0	NaN	NaN	8.0	NaN	NaN	280790
9	2011-11-01 01:00:00	NaN	0.6	NaN	NaN	63.0	71.0	NaN	33.0	NaN	6.0	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	
209908	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	3.0	19.0	41.0	NaN	NaN	5.0	NaN	NaN	28
209909	2011-09-01 00:00:00	0.2	0.3	0.2	NaN	1.0	18.0	42.0	14.0	NaN	5.0	NaN	0.6	28
209910	2011-09-01 00:00:00	0.7	0.1	1.1	0.04	1.0	12.0	46.0	8.0	5.0	5.0	1.25	0.9	28
209911	2011-09-01 00:00:00	NaN	NaN	NaN	0.18	4.0	27.0	42.0	NaN	NaN	NaN	1.28	NaN	28
209912	2011-09-01 00:00:00	NaN	0.2	NaN	NaN	1.0	24.0	45.0	NaN	NaN	6.0	NaN	NaN	28
209913	2011-09-01 00:00:00	NaN	0.2	NaN	NaN	8.0	37.0	NaN	14.0	NaN	6.0	NaN	NaN	28
209914	2011-09-01 00:00:00	0.3	NaN	0.7	NaN	12.0	33.0	NaN	16.0	9.0	5.0	NaN	1.4	28
209915	2011-09-01 00:00:00	NaN	0.2	NaN	NaN	8.0	29.0	35.0	NaN	NaN	NaN	NaN	NaN	28
209916	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	5.0	28.0	NaN	12.0	NaN	5.0	NaN	NaN	28
209917	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	1.0	17.0	NaN	11.0	7.0	NaN	NaN	NaN	28
209918	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	4.0	33.0	NaN	19.0	10.0	NaN	NaN	NaN	28
209919	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	3.0	20.0	39.0	NaN	NaN	NaN	NaN	NaN	28
209920	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	21.0	40.0	NaN	17.0	10.0	NaN	NaN	NaN	28
209921	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	9.0	20.0	43.0	NaN	NaN	NaN	NaN	NaN	28
209922	2011-09-01 00:00:00	0.3	NaN	0.3	0.11	3.0	32.0	NaN	20.0	NaN	NaN	1.34	2.0	28
209923	2011-09-01 00:00:00	NaN	0.2	NaN	NaN	5.0	19.0	44.0	NaN	NaN	NaN	NaN	NaN	28
209924	2011-09-01 00:00:00	NaN	0.1	NaN	NaN	6.0	29.0	NaN	11.0	NaN	7.0	NaN	NaN	28

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
209925	2011-09-01 00:00:00	NaN	NaN	NaN	0.23	1.0	21.0	28.0	NaN	NaN	NaN	1.44	NaN	28
209926	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	3.0	15.0	48.0	NaN	NaN	NaN	NaN	NaN	28
209927	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	4.0	33.0	38.0	13.0	NaN	NaN	NaN	NaN	28

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

Out[5]:

	BEN	CO	EBE	NMHC	NO	NO_2	
count	51393.000000	87127.000000	51350.000000	43517.000000	208954.000000	208973.000000	1
mean	0.815311	0.367868	0.970255	0.193821	28.146736	44.882243	
std	0.965508	0.281651	1.086046	0.097582	56.213615	32.495967	
min	0.100000	0.100000	0.100000	0.000000	1.000000	1.000000	
25%	0.200000	0.200000	0.300000	0.130000	3.000000	21.000000	
50%	0.400000	0.300000	0.600000	0.180000	8.000000	38.000000	
75%	1.000000	0.400000	1.300000	0.250000	25.000000	61.000000	
max	11.300000	4.400000	15.000000	4.810000	1126.000000	408.000000	

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

Out[6]:
```
(209928, 14)
```

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

Out[7]:
```
2938992
```

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	False	False	False	False	False	False	False	False	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
209923	False	True	False	True	True	False	False	False	True	True	True	True	True
209924	False	True	False	True	True	False	False	True	False	True	False	True	True
209925	False	True	True	True	False	False	False	False	True	True	True	False	True
209926	False	True	True	True	True	False	False	False	True	True	True	True	True
209927	False	True	True	True	True	False	False	False	False	True	True	True	True

209928 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	s
1	2011-11-01 01:00:00	2.5	0.4	3.5	0.26	68.0	92.0	3.0	40.0	24.0	9.0	1.54	8.7	280
6	2011-11-01 01:00:00	0.7	0.3	1.1	0.16	17.0	66.0	7.0	22.0	16.0	2.0	1.36	1.7	280
25	2011-11-01 02:00:00	1.8	0.3	2.8	0.20	34.0	76.0	3.0	34.0	21.0	8.0	1.71	7.4	280
30	2011-11-01 02:00:00	1.0	0.4	1.3	0.18	31.0	67.0	5.0	25.0	18.0	3.0	1.40	2.9	280
49	2011-11-01 03:00:00	1.3	0.2	2.4	0.22	29.0	72.0	3.0	33.0	20.0	8.0	1.75	6.2	280
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
209862	2011-08-31 22:00:00	0.4	0.1	1.0	0.06	1.0	13.0	33.0	21.0	6.0	5.0	1.26	0.7	280
209881	2011-08-31 23:00:00	0.9	0.1	1.8	0.16	11.0	45.0	30.0	32.0	17.0	3.0	1.34	4.9	280
209886	2011-08-31 23:00:00	0.6	0.1	1.1	0.05	1.0	12.0	48.0	19.0	7.0	5.0	1.26	0.9	280
209905	2011-09-01 00:00:00	0.6	0.1	1.3	0.15	6.0	35.0	34.0	21.0	12.0	3.0	1.32	3.8	280
209910	2011-09-01 00:00:00	0.7	0.1	1.1	0.04	1.0	12.0	46.0	8.0	5.0	5.0	1.25	0.9	280

16460 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']]
```

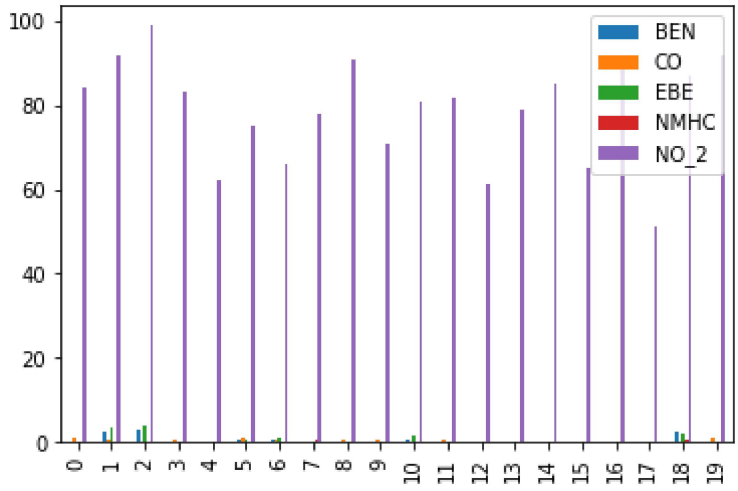
In [12]:
```
dd=sd.head(20)
dd
```

Out[12]:

	BEN	CO	EBE	NMHC	NO_2
0	NaN	1.0	NaN	NaN	84.0
1	2.5	0.4	3.5	0.26	92.0
2	2.9	NaN	3.8	NaN	99.0
3	NaN	0.6	NaN	NaN	83.0
4	NaN	NaN	NaN	NaN	62.0
5	0.5	0.8	0.3	NaN	75.0
6	0.7	0.3	1.1	0.16	66.0
7	NaN	NaN	NaN	0.36	78.0
8	NaN	0.7	NaN	NaN	91.0
9	NaN	0.6	NaN	NaN	71.0
10	0.3	NaN	1.4	NaN	81.0
11	NaN	0.6	NaN	NaN	82.0
12	NaN	NaN	NaN	NaN	61.0
13	NaN	NaN	NaN	NaN	79.0
14	NaN	NaN	NaN	NaN	85.0
15	NaN	NaN	NaN	NaN	65.0
16	NaN	NaN	NaN	NaN	90.0
17	NaN	NaN	NaN	NaN	51.0
18	2.3	NaN	1.9	0.27	87.0
19	NaN	0.8	NaN	NaN	92.0

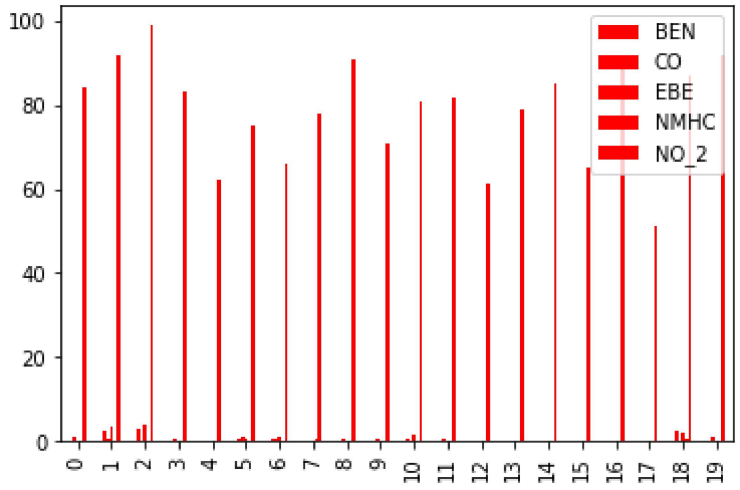
In [13]:
```
dd.plot.bar()
```

Out[13]:
```
<AxesSubplot:>
```



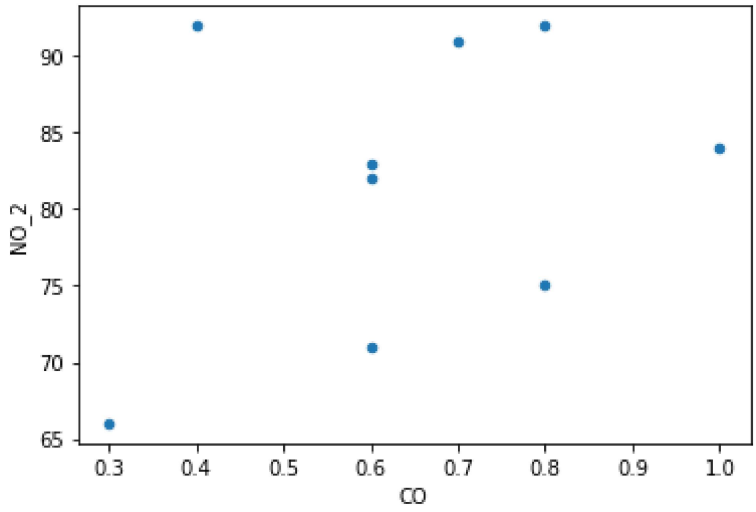
In [14]:
```
dd.plot.bar(color='r')
```

Out[14]: <AxesSubplot:>



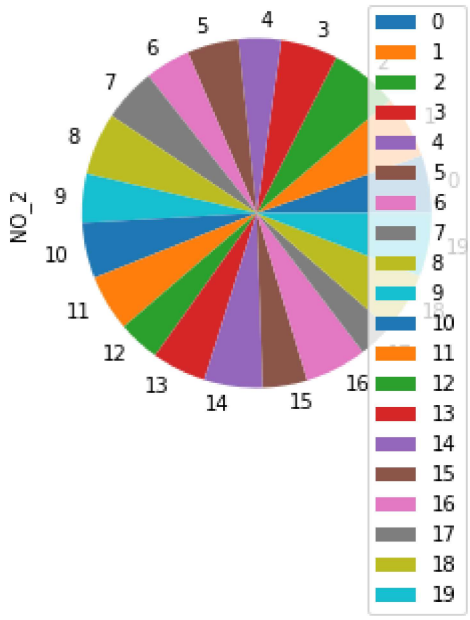
In [15]:
```
dd.plot.scatter(x='CO',y='NO_2')
```

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



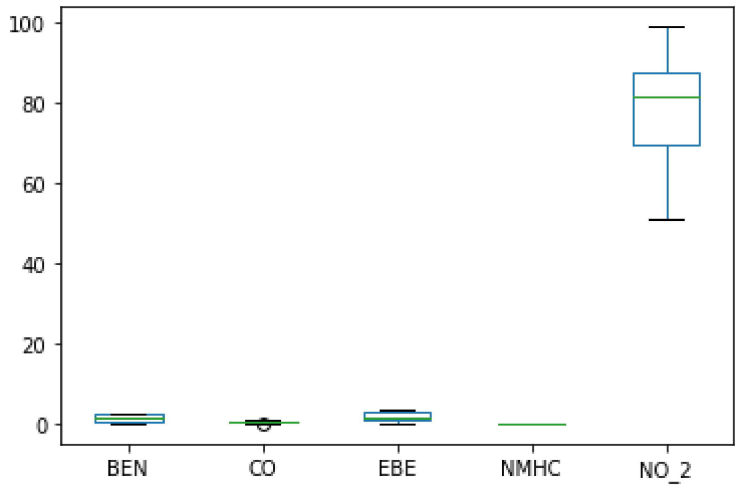
In [16]:
```
dd.plot.pie(y='NO_2')
```

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



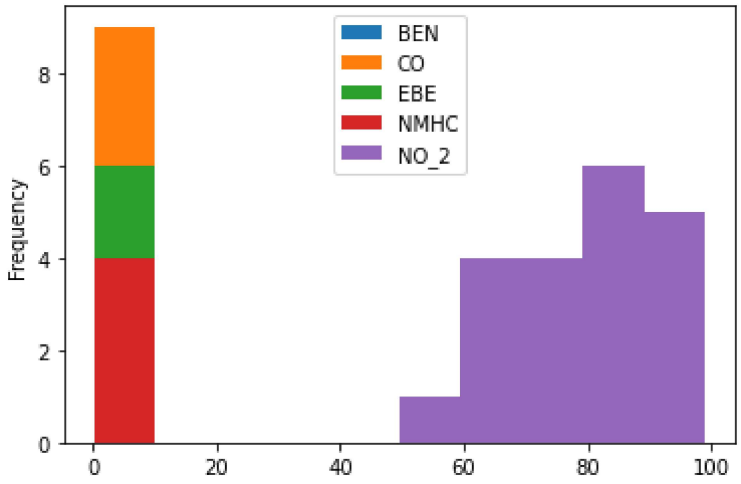
In [17]:
```
dd.plot.box()
```

Out[17]: <AxesSubplot:>



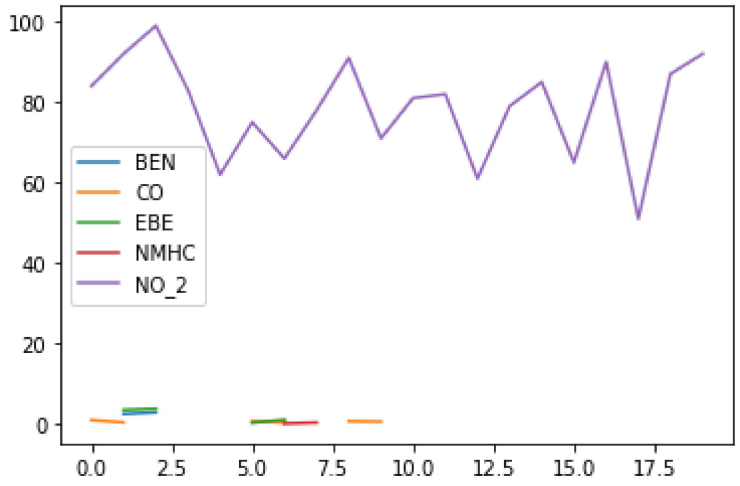
In [18]:
```
dd.plot.hist()
```

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



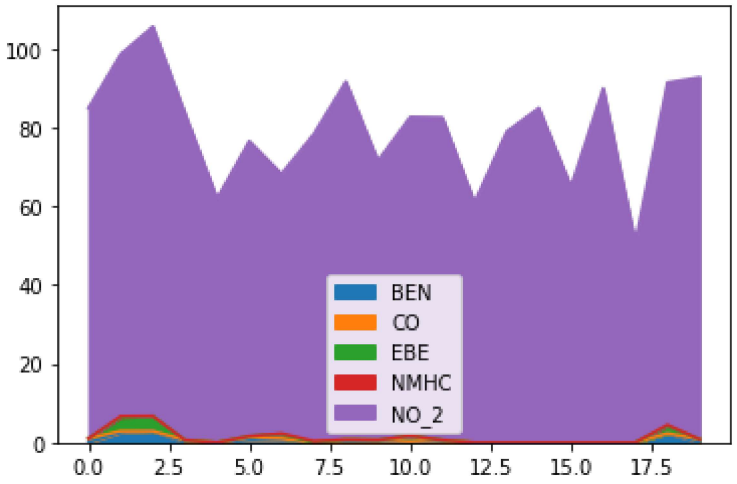
In [19]:
```
dd.plot.line()
```

Out[19]: <AxesSubplot:>



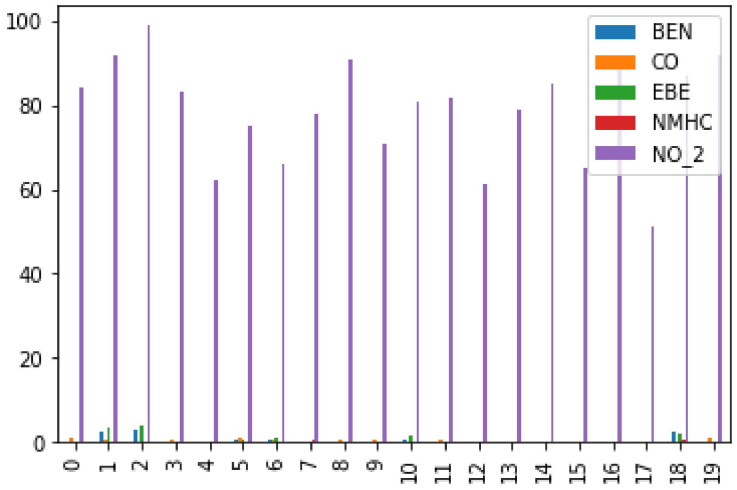
In [20]:
```
dd.plot.area()
```

Out[20]: <AxesSubplot:>



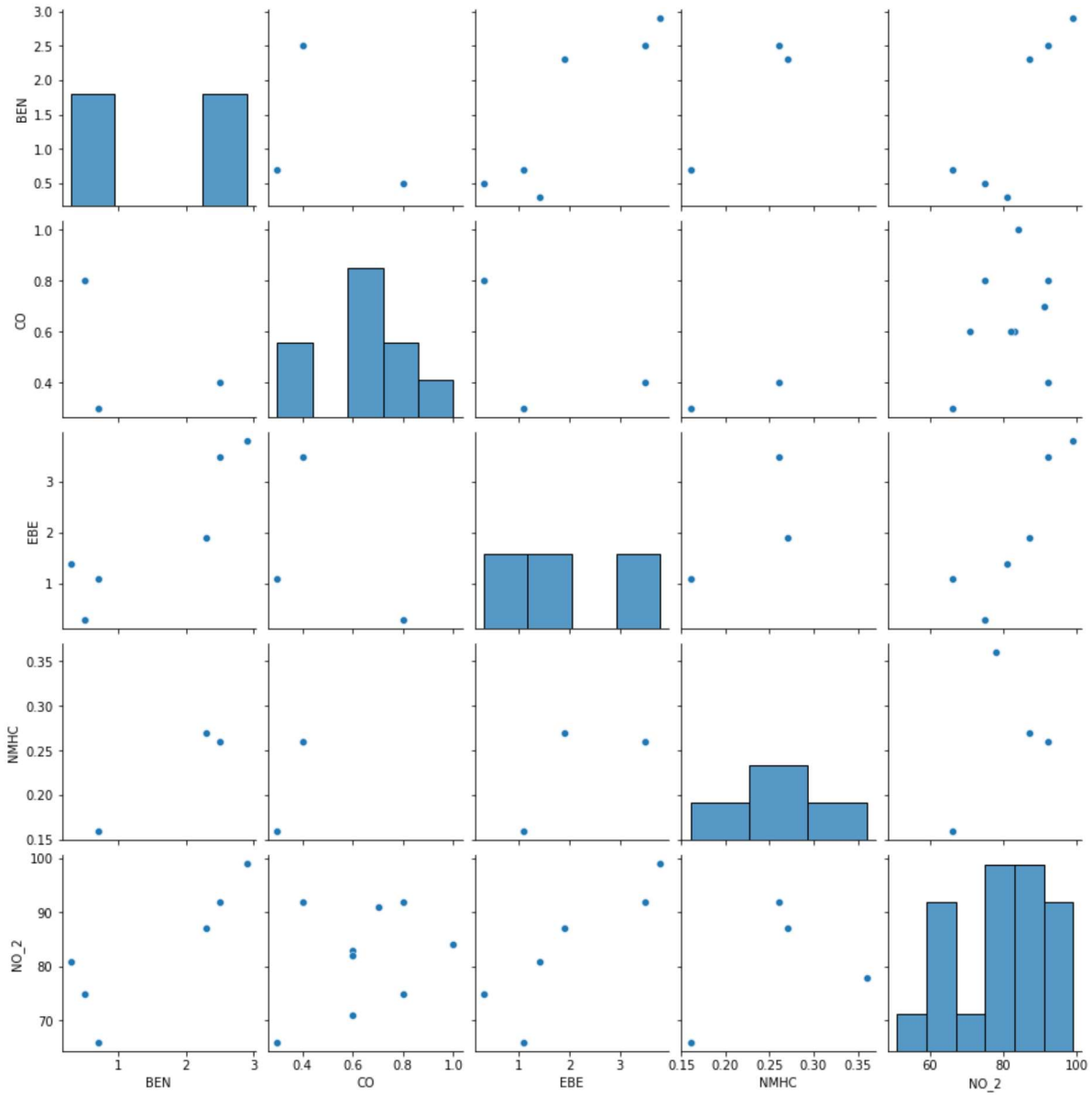
In [21]:
```
dd.plot.bar()
```

Out[21]: <AxesSubplot:>



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

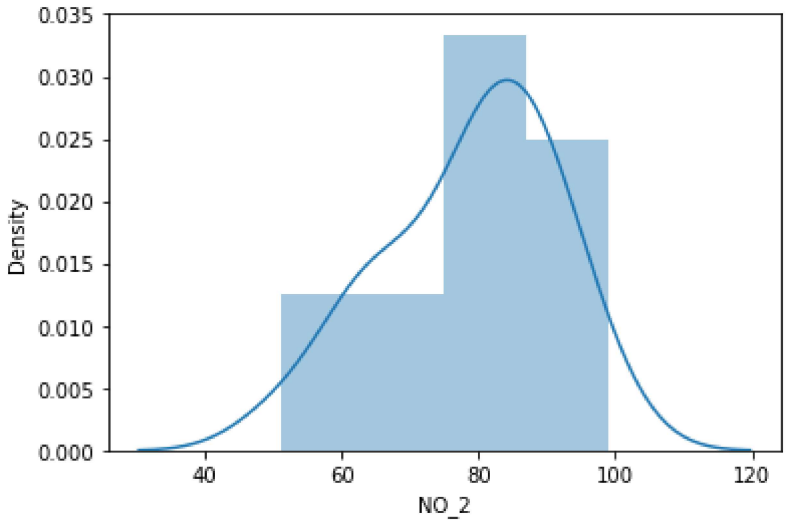
Out[22]:
```
<seaborn.axisgrid.PairGrid at 0x193ef976cd0>
```



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'>
```



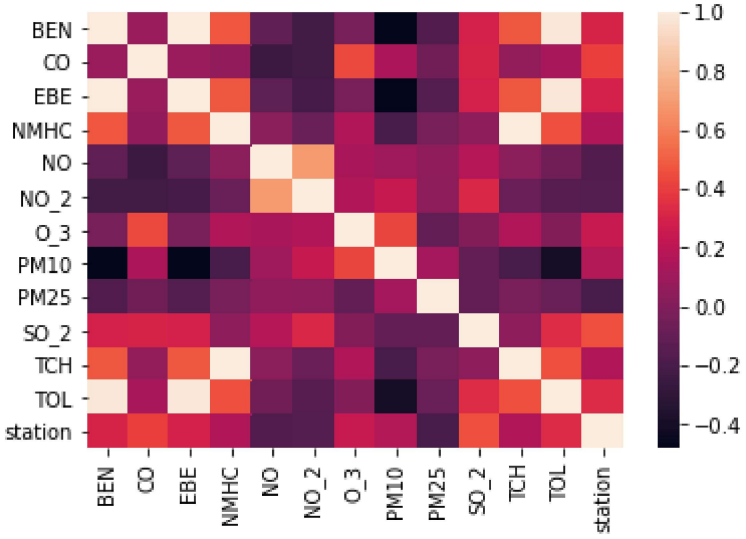
In [24]:
```
ds=data.fillna(20)
```

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_)
```

```
28079024.239080243
```

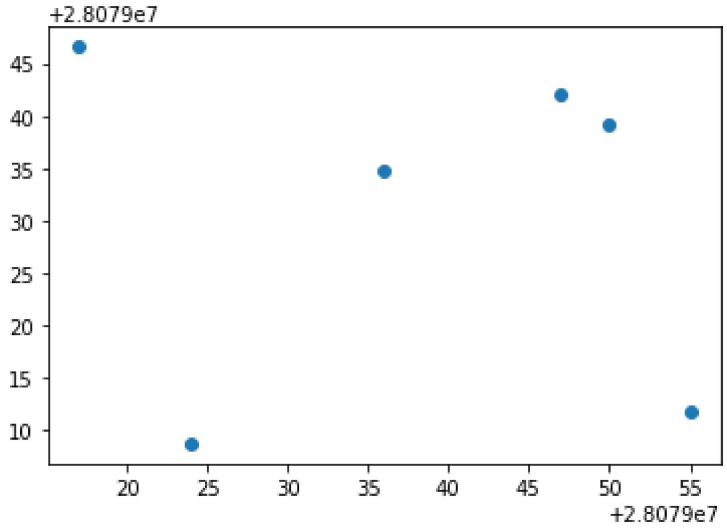
In [32]:
```
coeff= pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[32]:

	Co-efficient
BEN	-2.046027
CO	0.487406
EBE	2.861552
NMHC	0.636439
NO_2	-0.264777

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

Out[33]:
```
<matplotlib.collections.PathCollection at 0x193f2635e50>
```



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

```
-1.6962359363999515
```

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

Out[35]:
```
-1.6962359363999515
```

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

Out[36]:
```
0.4453702958576773
```

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]:
```
-1.4253625975304316
```

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

Out[40]:
```
0.44425698909440503
```

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]:
```
-1.00815353490863
```

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

Out[43]:
```
0.42952965002635835
```

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_)
```

```
[ 0.26269544  0.51638978  0.4739114   0.60125289 -0.23748442]
```

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

```
28079024.00224517
```

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

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

```
-1.4097436233275866
```

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)
```

```
[28079050]
```

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]:
```
0.004028609623069467
```

In [62]:
```
ged=data[['BEN','CO','EBE','NMHC','NO_2','O_3','PM10','SO_2','TCH','TOL','stati
```

In [63]:
```
d=ged.fillna(20)
```

In [64]:
```
dg=d.head(100)
```

In [65]:
```
x=dg[['BEN','CO','EBE','NMHC','NO_2','O_3','PM10','SO_2','TCH','TOL']]
y=dg['station']
```

In [66]:
```
print(len(x))
print(len(y))
```

```
100
100
```

In [67]:
```
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 [68]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[68]:
```
RandomForestClassifier()
```

In [69]:
```
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 [70]:
```
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[70]:
```
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')
```

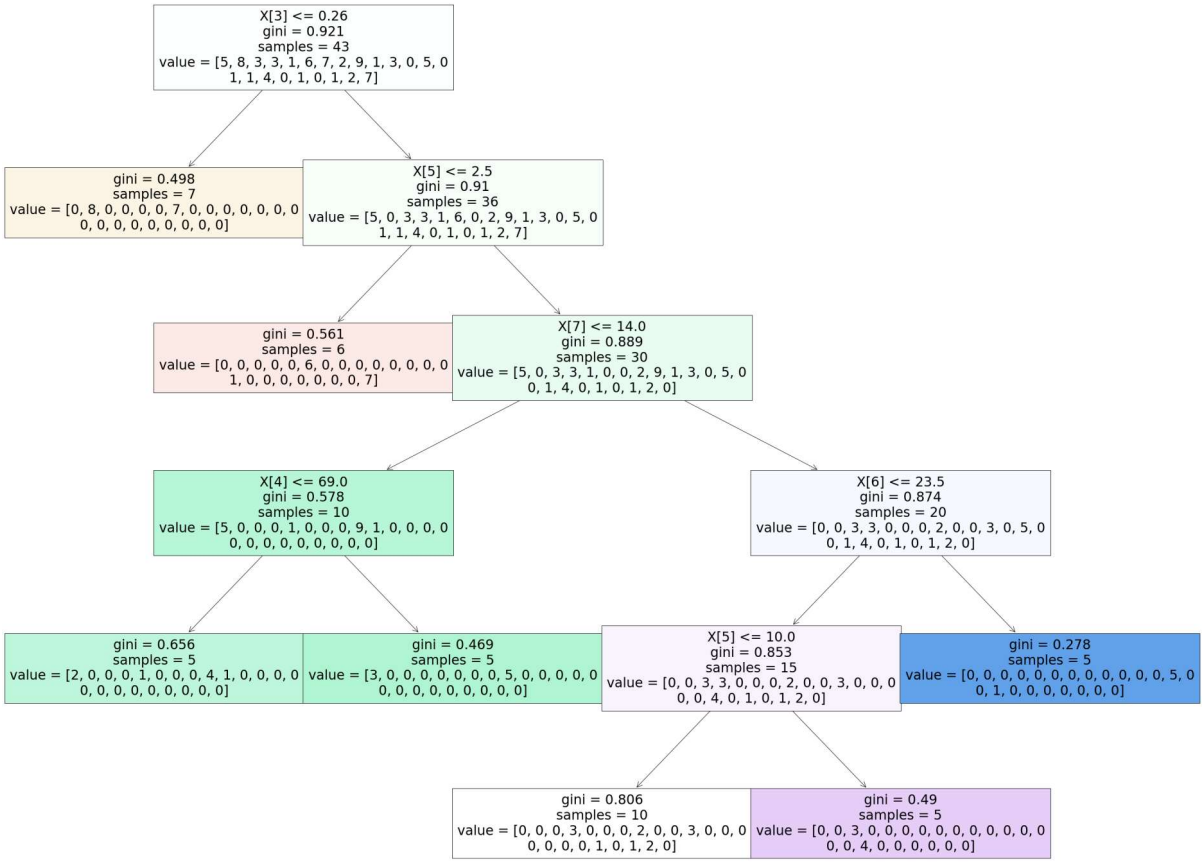
In [71]:
```
grid_search.best_score_
```

Out[71]:
```
0.6
```

In [72]:
```
rfc_best=grid_search.best_estimator_
```

In [73]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(50,40))
plot_tree(rfc_best.estimators_[5],filled=True)
```

Out[73]:
```
[Text(697.5, 1993.2, 'X[3] <= 0.26\ngini = 0.921\nsamples = 43\nvalue = [5,
8, 3, 3, 1, 6, 7, 2, 9, 1, 3, 0, 5, 0\n1, 1, 4, 0, 1, 0, 1, 2, 7]'),
 Text(348.75, 1630.8000000000002, 'gini = 0.498\nsamples = 7\nvalue = [0, 8,
0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1046.25, 1630.8000000000002, 'X[5] <= 2.5\ngini = 0.91\nsamples = 36\nv
alue = [5, 0, 3, 3, 1, 6, 0, 2, 9, 1, 3, 0, 5, 0\n1, 1, 4, 0, 1, 0, 1, 2,
7]'),
 Text(697.5, 1268.4, 'gini = 0.561\nsamples = 6\nvalue = [0, 0, 0, 0, 0, 6,
0, 0, 0, 0, 0, 0, 0, 0\n1, 0, 0, 0, 0, 0, 0, 0, 7]'),
 Text(1395.0, 1268.4, 'X[7] <= 14.0\ngini = 0.889\nsamples = 30\nvalue = [5,
0, 3, 3, 1, 0, 0, 2, 9, 1, 3, 0, 5, 0\n0, 1, 4, 0, 1, 0, 1, 2, 0]'),
 Text(697.5, 906.0, 'X[4] <= 69.0\ngini = 0.578\nsamples = 10\nvalue = [5, 0,
0, 0, 1, 0, 0, 0, 9, 1, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(348.75, 543.5999999999999, 'gini = 0.656\nsamples = 5\nvalue = [2, 0,
0, 0, 1, 0, 0, 0, 4, 1, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1046.25, 543.5999999999999, 'gini = 0.469\nsamples = 5\nvalue = [3, 0,
0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(2092.5, 906.0, 'X[6] <= 23.5\ngini = 0.874\nsamples = 20\nvalue = [0,
0, 3, 3, 0, 0, 0, 2, 0, 0, 3, 0, 5, 0\n0, 1, 4, 0, 1, 0, 1, 2, 0]'),
 Text(1743.75, 543.5999999999999, 'X[5] <= 10.0\ngini = 0.853\nsamples = 15\n
value = [0, 0, 3, 3, 0, 0, 0, 2, 0, 0, 3, 0, 0, 0\n0, 0, 4, 0, 1, 0, 1, 2,
0]'),
 Text(1395.0, 181.19999999999982, 'gini = 0.806\nsamples = 10\nvalue = [0, 0,
0, 3, 0, 0, 0, 2, 0, 0, 3, 0, 0, 0\n0, 0, 0, 0, 1, 0, 1, 2, 0]'),
 Text(2092.5, 181.19999999999982, 'gini = 0.49\nsamples = 5\nvalue = [0, 0,
3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 4, 0, 0, 0, 0, 0, 0]'),
 Text(2441.25, 543.5999999999999, 'gini = 0.278\nsamples = 5\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0\n0, 1, 0, 0, 0, 0, 0, 0, 0]')]
```



Conclusion : LogisticRegression() [28079050] HIGH RANGE

In [ ]: