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_2018.csv")
data
```

Out[2]:

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2	T
0	2018-03-01 01:00:00	NaN	NaN	0.3	NaN	NaN	1.0	29.0	31.0	NaN	NaN	NaN	2.0	N
1	2018-03-01 01:00:00	0.5	1.39	0.3	0.2	0.02	6.0	40.0	49.0	52.0	5.0	4.0	3.0	1
2	2018-03-01 01:00:00	0.4	NaN	NaN	0.2	NaN	4.0	41.0	47.0	NaN	NaN	NaN	NaN	N
3	2018-03-01 01:00:00	NaN	NaN	0.3	NaN	NaN	1.0	35.0	37.0	54.0	NaN	NaN	NaN	N
4	2018-03-01 01:00:00	NaN	NaN	NaN	NaN	NaN	1.0	27.0	29.0	49.0	NaN	NaN	3.0	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
69091	2018-02-01 00:00:00	NaN	NaN	0.5	NaN	NaN	66.0	91.0	192.0	1.0	35.0	22.0	NaN	N
69092	2018-02-01 00:00:00	NaN	NaN	0.7	NaN	NaN	87.0	107.0	241.0	NaN	29.0	NaN	15.0	N
69093	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	28.0	48.0	91.0	2.0	NaN	NaN	NaN	N
69094	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	141.0	103.0	320.0	2.0	NaN	NaN	NaN	N
69095	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	69.0	96.0	202.0	3.0	26.0	NaN	NaN	N

69096 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	2018-03-01 01:00:00	NaN	NaN	0.3	NaN	NaN	1.0	29.0	31.0	NaN	NaN	NaN	2.0	NaN	Na
1	2018-03-01 01:00:00	0.5	1.39	0.3	0.2	0.02	6.0	40.0	49.0	52.0	5.0	4.0	3.0	1.41	0
2	2018-03-01 01:00:00	0.4	NaN	NaN	0.2	NaN	4.0	41.0	47.0	NaN	NaN	NaN	NaN	NaN	1
3	2018-03-01 01:00:00	NaN	NaN	0.3	NaN	NaN	1.0	35.0	37.0	54.0	NaN	NaN	NaN	NaN	Na
4	2018-03-01 01:00:00	NaN	NaN	NaN	NaN	NaN	1.0	27.0	29.0	49.0	NaN	NaN	3.0	NaN	Na
5	2018-03-01 01:00:00	0.3	NaN	0.3	0.2	NaN	1.0	27.0	29.0	57.0	8.0	NaN	6.0	NaN	1
6	2018-03-01 01:00:00	0.4	1.11	0.2	0.1	0.06	1.0	25.0	27.0	55.0	5.0	4.0	4.0	1.16	1
7	2018-03-01 01:00:00	NaN	NaN	NaN	NaN	NaN	1.0	37.0	39.0	54.0	NaN	NaN	NaN	NaN	Na
8	2018-03-01 01:00:00	NaN	NaN	0.5	NaN	NaN	3.0	43.0	47.0	29.0	NaN	NaN	5.0	NaN	Na
9	2018-03-01 01:00:00	NaN	NaN	0.2	NaN	NaN	2.0	26.0	29.0	NaN	4.0	NaN	6.0	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	T
69076	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	226.0	124.0	471.0	1.0	NaN	NaN	12.0	N
69077	2018-02-01 00:00:00	1.1	NaN	0.6	0.8	NaN	87.0	93.0	227.0	1.0	32.0	NaN	8.0	N
69078	2018-02-01 00:00:00	1.3	1.14	0.4	0.8	0.10	54.0	73.0	155.0	1.0	27.0	16.0	5.0	1
69079	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	64.0	83.0	182.0	3.0	NaN	NaN	NaN	N
69080	2018-02-01 00:00:00	NaN	NaN	0.5	NaN	NaN	117.0	90.0	269.0	5.0	NaN	NaN	11.0	N
69081	2018-02-01 00:00:00	NaN	NaN	1.3	NaN	NaN	303.0	158.0	623.0	NaN	64.0	NaN	25.0	N
69082	2018-02-01 00:00:00	2.0	NaN	NaN	1.6	NaN	68.0	99.0	204.0	NaN	30.0	20.0	7.0	N
69083	2018-02-01 00:00:00	NaN	NaN	0.9	NaN	NaN	144.0	111.0	331.0	1.0	NaN	NaN	NaN	N
69084	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	221.0	141.0	480.0	NaN	64.0	NaN	15.0	N
69085	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	111.0	94.0	264.0	NaN	41.0	29.0	NaN	N
69086	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	75.0	102.0	217.0	NaN	31.0	20.0	NaN	N
69087	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	145.0	102.0	325.0	3.0	NaN	NaN	NaN	N
69088	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	92.0	109.0	250.0	NaN	31.0	21.0	NaN	N
69089	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	258.0	145.0	541.0	2.0	NaN	NaN	NaN	N
69090	2018-02-01 00:00:00	1.3	1.55	NaN	1.2	0.13	63.0	94.0	190.0	NaN	35.0	NaN	NaN	1
69091	2018-02-01 00:00:00	NaN	NaN	0.5	NaN	NaN	66.0	91.0	192.0	1.0	35.0	22.0	NaN	N
69092	2018-02-01 00:00:00	NaN	NaN	0.7	NaN	NaN	87.0	107.0	241.0	NaN	29.0	NaN	15.0	N

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2	T
69093	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	28.0	48.0	91.0	2.0	NaN	NaN	NaN	N
69094	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	141.0	103.0	320.0	2.0	NaN	NaN	NaN	N
69095	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	69.0	96.0	202.0	3.0	26.0	NaN	NaN	N

In [5]:

```
data.describe()
```

Out[5]:

	BEN	CH4	CO	EBE	NMHC	NO	
count	16950.000000	8440.000000	28598.000000	16949.000000	8440.000000	68826.000000	68826.
mean	0.555864	1.285379	0.344433	0.300531	0.065256	19.893253	38.
std	0.455012	0.187705	0.202271	0.402112	0.041480	40.641962	28
min	0.100000	0.020000	0.100000	0.100000	0.000000	1.000000	1.
25%	0.300000	1.140000	0.200000	0.100000	0.040000	1.000000	16.
50%	0.400000	1.230000	0.300000	0.200000	0.060000	5.000000	32.
75%	0.700000	1.400000	0.400000	0.400000	0.080000	18.000000	55.
max	8.400000	3.920000	3.200000	14.900000	0.490000	774.000000	276.

In [6]:

```
np.shape(data)
```

Out[6]:

```
(69096, 16)
```

In [7]:

```
np.size(data)
```

Out[7]:

```
1105536
```

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	False	True	True	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	True	True	False	True	False	False	False	True	True	True	True
3	False	True	True	False	True	True	False	False	False	False	True	True	True
4	False	True	True	True	True	True	False	False	False	False	True	True	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
69091	False	True	True	False	True	True	False	False	False	False	False	False	True
69092	False	True	True	False	True	True	False	False	False	True	False	True	False
69093	False	True	True	True	True	True	False	False	False	False	True	True	True
69094	False	True	True	True	True	True	False	False	False	False	True	True	True
69095	False	True	True	True	True	True	False	False	False	False	False	True	True

69096 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	TC
1	2018-03-01 01:00:00	0.5	1.39	0.3	0.2	0.02	6.0	40.0	49.0	52.0	5.0	4.0	3.0	1.4
6	2018-03-01 01:00:00	0.4	1.11	0.2	0.1	0.06	1.0	25.0	27.0	55.0	5.0	4.0	4.0	1.1
25	2018-03-01 02:00:00	0.4	1.42	0.2	0.1	0.01	4.0	26.0	32.0	64.0	4.0	4.0	3.0	1.4
30	2018-03-01 02:00:00	0.3	1.10	0.2	0.1	0.05	1.0	12.0	13.0	69.0	5.0	4.0	4.0	1.1
49	2018-03-01 03:00:00	0.3	1.41	0.2	0.1	0.01	3.0	16.0	20.0	68.0	3.0	2.0	3.0	1.4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
69030	2018-01-31 22:00:00	1.8	1.21	0.7	1.7	0.19	151.0	129.0	361.0	1.0	45.0	26.0	11.0	1.4
69049	2018-01-31 23:00:00	3.1	1.87	1.2	2.0	0.35	296.0	162.0	615.0	3.0	39.0	23.0	8.0	2.2
69054	2018-01-31 23:00:00	1.6	1.17	0.6	1.4	0.15	127.0	106.0	301.0	1.0	43.0	25.0	8.0	1.3
69073	2018-02-01 00:00:00	3.2	1.53	1.0	2.1	0.19	125.0	117.0	309.0	3.0	37.0	24.0	6.0	1.7
69078	2018-02-01 00:00:00	1.3	1.14	0.4	0.8	0.10	54.0	73.0	155.0	1.0	27.0	16.0	5.0	1.2

4562 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]:

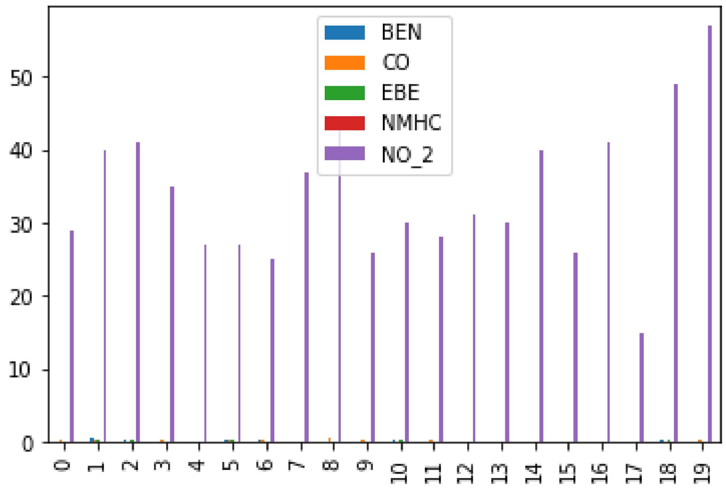
	BEN	CO	EBE	NMHC	NO_2
0	NaN	0.3	NaN	NaN	29.0
1	0.5	0.3	0.2	0.02	40.0
2	0.4	NaN	0.2	NaN	41.0
3	NaN	0.3	NaN	NaN	35.0
4	NaN	NaN	NaN	NaN	27.0
5	0.3	0.3	0.2	NaN	27.0
6	0.4	0.2	0.1	0.06	25.0
7	NaN	NaN	NaN	NaN	37.0
8	NaN	0.5	NaN	NaN	43.0
9	NaN	0.2	NaN	NaN	26.0
10	0.4	NaN	0.3	NaN	30.0
11	NaN	0.3	NaN	NaN	28.0
12	NaN	NaN	NaN	NaN	31.0
13	NaN	NaN	NaN	NaN	30.0
14	NaN	NaN	NaN	NaN	40.0
15	NaN	NaN	NaN	NaN	26.0
16	NaN	NaN	NaN	NaN	41.0
17	NaN	NaN	NaN	NaN	15.0
18	0.3	NaN	0.3	0.03	49.0
19	NaN	0.2	NaN	NaN	57.0

In [13]:

```
dd.plot.bar()
```

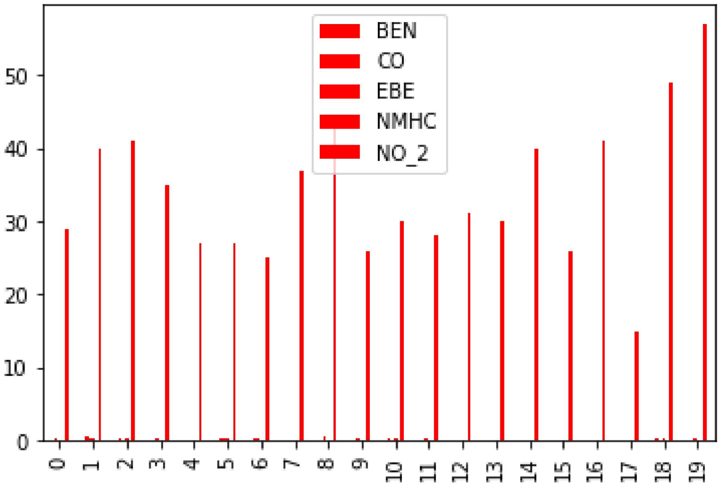
Out[13]:

```
<AxesSubplot:>
```



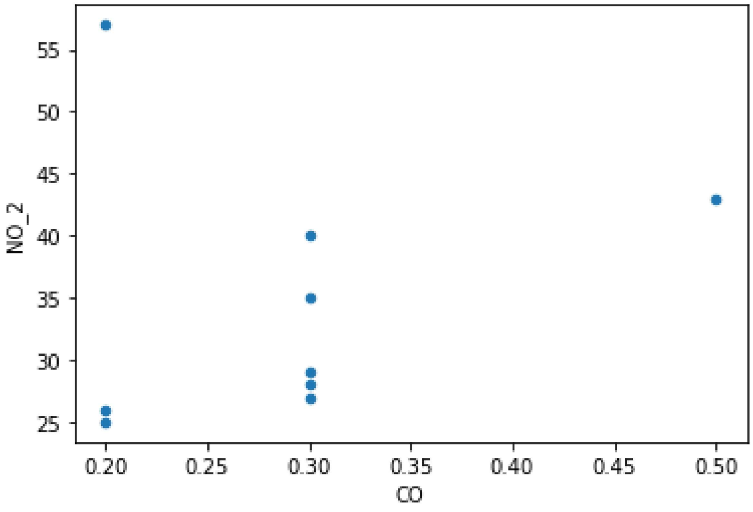
```
In [14]: dd.plot.bar(color='r')
```

```
Out[14]: <AxesSubplot:>
```



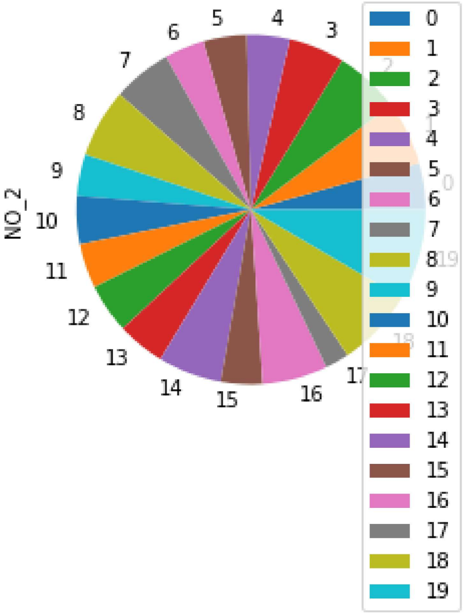
```
In [15]: dd.plot.scatter(x='CO',y='NO_2')
```

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



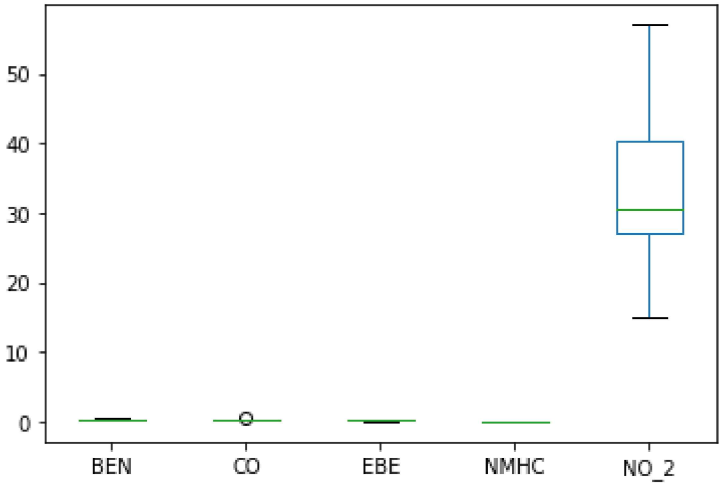
```
In [16]: dd.plot.pie(y='NO_2')
```

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



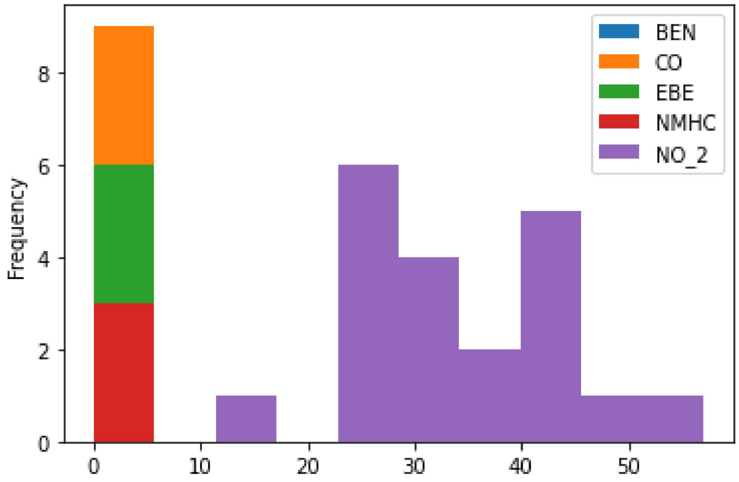
```
In [17]: dd.plot.box()
```

```
Out[17]: <AxesSubplot:>
```



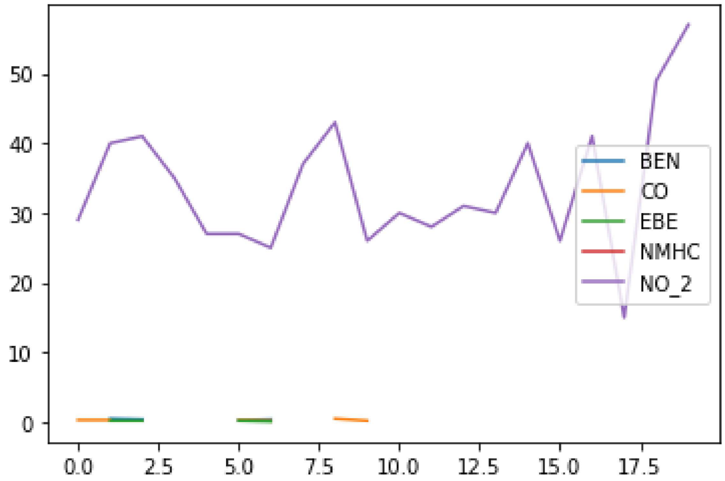
```
In [18]: dd.plot.hist()
```

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



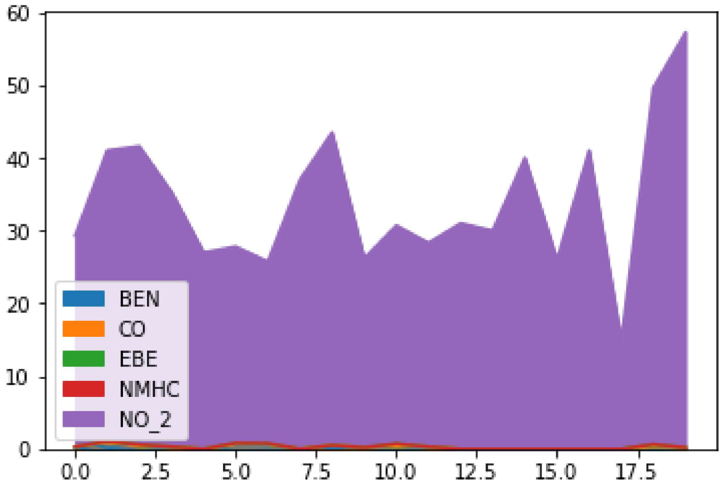
```
In [19]: dd.plot.line()
```

```
Out[19]: <AxesSubplot:>
```



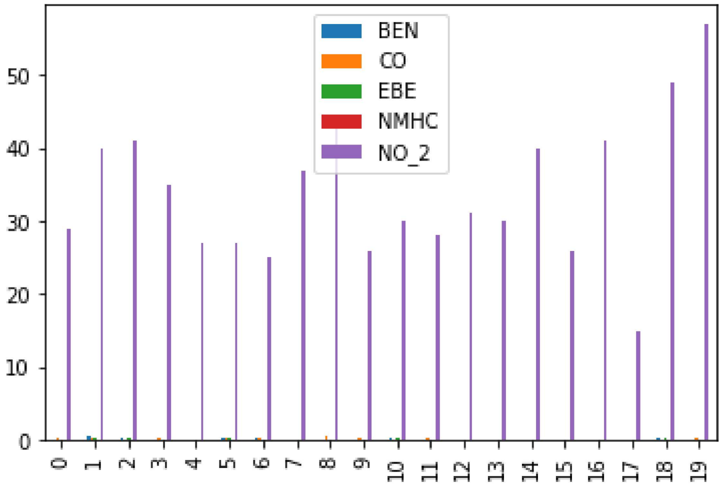
In [20]:
```
dd.plot.area()
```

Out[20]:
```
<AxesSubplot:>
```



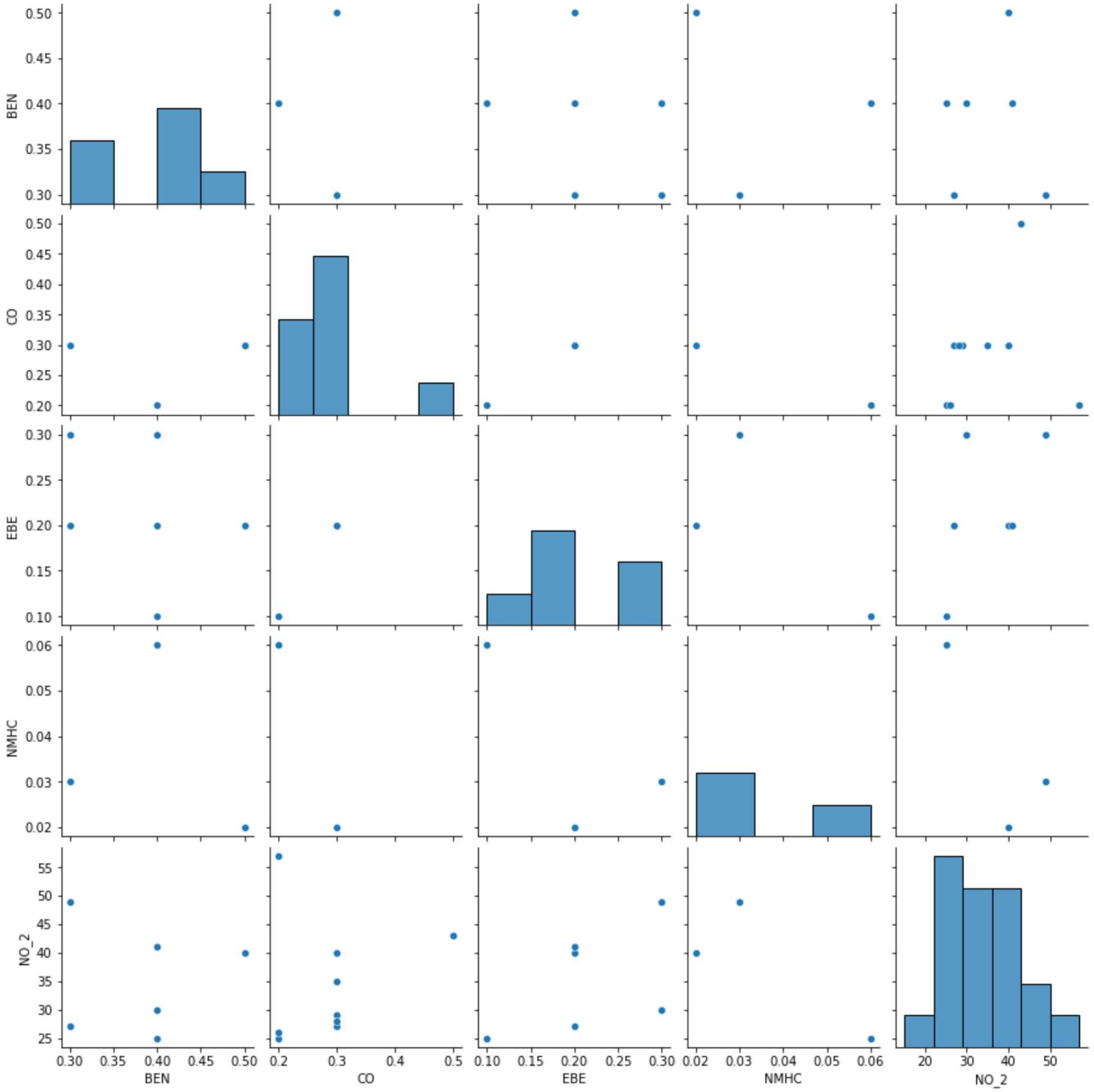
In [21]:
```
dd.plot.bar()
```

Out[21]:
```
<AxesSubplot:>
```



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

```
Out[22]: <seaborn.axisgrid.PairGrid at 0x11cfdfa56d0>
```



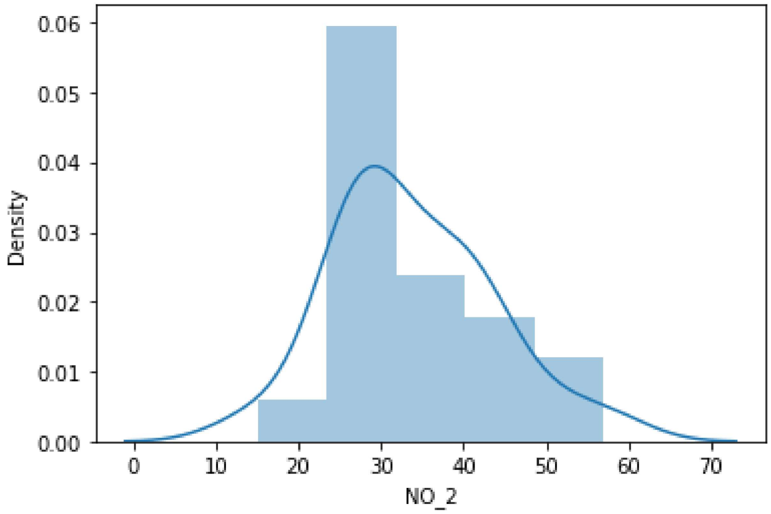
In [23]:

```
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]:

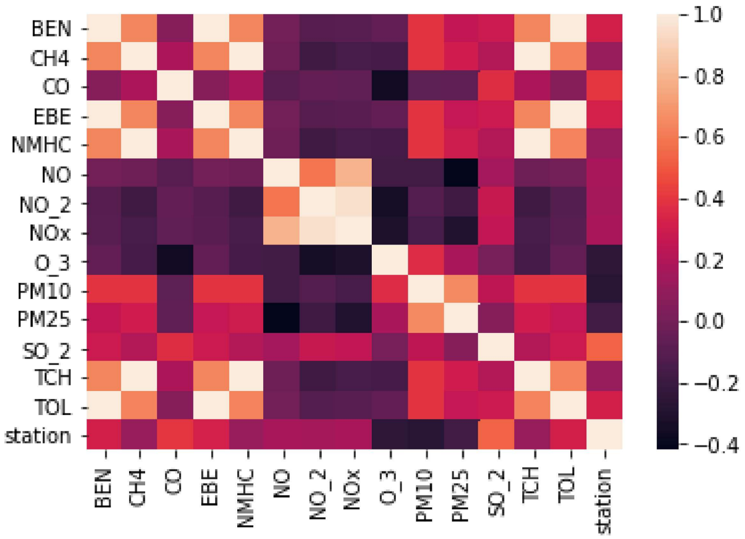
```
sd1=ssd[['BEN','CO', 'EBE', 'NMHC', 'NO_2']]
```

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_)
```

```
28079026.630780514
```

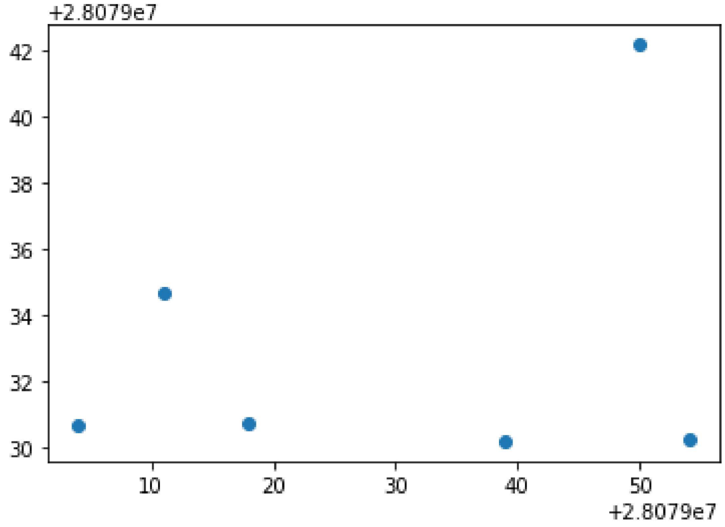
In [32]:
```
coeff= pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[32]:

	Co-efficient
BEN	-85.063811
CO	0.304875
EBE	84.583095
NMHC	0.013854
NO_2	0.457544

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

Out[33]:
```
<matplotlib.collections.PathCollection at 0x11c82793bb0>
```



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

         0.04462550703001378
```

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

Out[35]: 0.04462550703001378
```

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

Out[36]: 0.45268405168051273
```

```
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]: -0.19280497437524602
```

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

Out[40]: 0.3726204582781336
```

```
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]: -0.12159577377462316
```

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

Out[43]: 0.3531486978794881
```

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.         0.6224899  0.01065925 0.32510087 0.74088171]

In [46]: print(en.intercept_)

         28078996.72619016

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

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

         -0.19032204159505484

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]:

```
0.0003265926989572284
```

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]:

```
0.5857142857142856
```

In [70]:

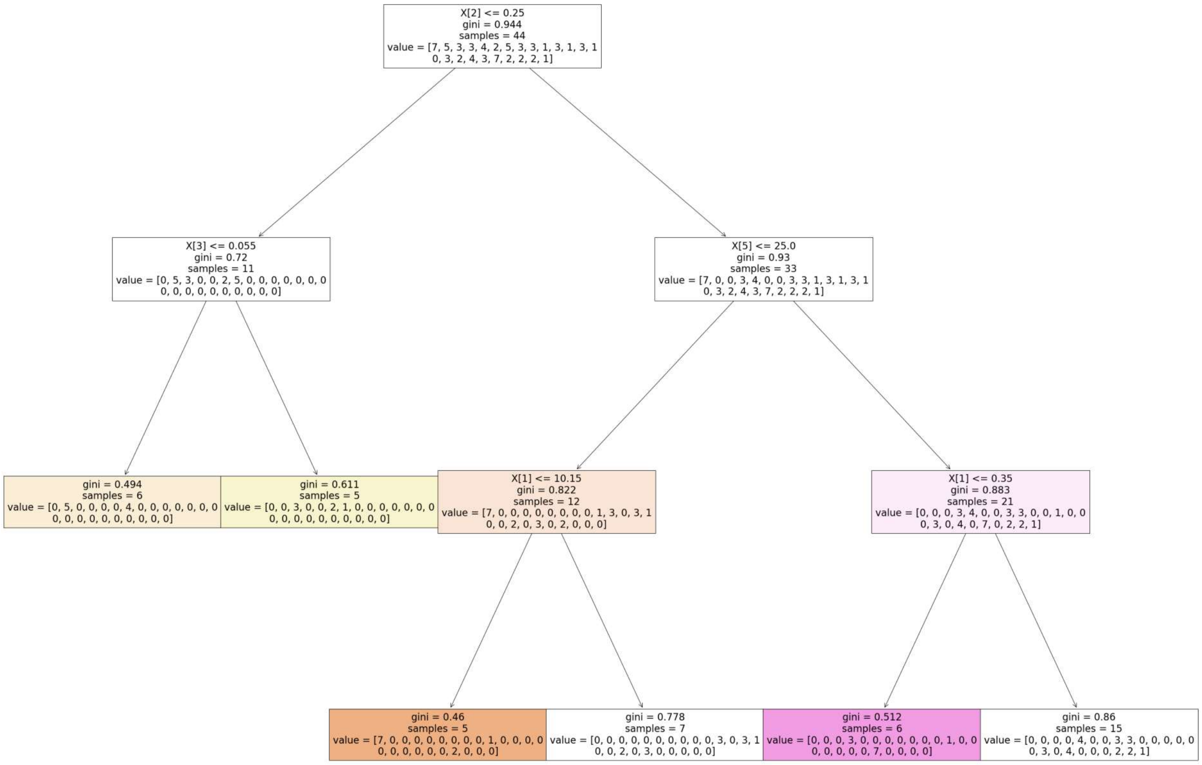
```
rfc_best=grid_search.best_estimator_
```

In [71]:

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

Out[71]:

```
[Text(1141.3636363636363, 1902.6000000000001, 'X[2] <= 0.25\ngini = 0.944\nsa
mples = 44\nvalue = [7, 5, 3, 3, 4, 2, 5, 3, 3, 1, 3, 1, 3, 1\n0, 3, 2, 4, 3,
7, 2, 2, 2, 1]'),
 Text(507.27272727272725, 1359.0, 'X[3] <= 0.055\ngini = 0.72\nsamples = 11\n
value = [0, 5, 3, 0, 0, 2, 5, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0]'),
 Text(253.63636363636363, 815.4000000000001, 'gini = 0.494\nsamples = 6\nvalu
e = [0, 5, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0]'),
 Text(760.9090909090909, 815.4000000000001, 'gini = 0.611\nsamples = 5\nvalue
= [0, 0, 3, 0, 0, 2, 1, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1775.4545454545455, 1359.0, 'X[5] <= 25.0\ngini = 0.93\nsamples = 33\nv
alue = [7, 0, 0, 3, 4, 0, 0, 3, 3, 1, 3, 1, 3, 1\n0, 3, 2, 4, 3, 7, 2, 2, 2,
1]'),
 Text(1268.181818181818, 815.4000000000001, 'X[1] <= 10.15\ngini = 0.822\nsam
ples = 12\nvalue = [7, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 0, 3, 1\n0, 0, 2, 0, 3,
0, 2, 0, 0, 0]'),
 Text(1014.5454545454545, 271.79999999999995, 'gini = 0.46\nsamples = 5\nvalu
e = [7, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0\n0, 0, 0, 0, 0, 2, 0, 0,
0]'),
 Text(1521.8181818181818, 271.79999999999995, 'gini = 0.778\nsamples = 7\nval
ue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 3, 1\n0, 0, 2, 0, 3, 0, 0, 0, 0,
0]'),
 Text(2282.7272727272725, 815.4000000000001, 'X[1] <= 0.35\ngini = 0.883\nsam
ples = 21\nvalue = [0, 0, 0, 3, 4, 0, 0, 3, 3, 0, 0, 1, 0, 0\n0, 3, 0, 4, 0,
7, 0, 2, 2, 1]'),
 Text(2029.090909090909, 271.79999999999995, 'gini = 0.512\nsamples = 6\nvalu
e = [0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0\n0, 0, 0, 0, 0, 7, 0, 0, 0,
0]'),
 Text(2536.363636363636, 271.79999999999995, 'gini = 0.86\nsamples = 15\nvalu
e = [0, 0, 0, 0, 4, 0, 0, 3, 3, 0, 0, 0, 0, 0\n0, 3, 0, 4, 0, 0, 0, 2, 2,
1]')]
```



Conclusion : LinearRegression() 28079026.630780514 HIGH RANGE

In [ ]: