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_2014.csv")
data
```

Out[2]:

| | date | BEN | CO | EBE | NMHC | NO | NO_2 | O_3 | PM10 | PM25 | SO_2 | TCH | TOL | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2014-06-01 01:00:00 | NaN | 0.2 | NaN | NaN | 3.0 | 10.0 | NaN | NaN | NaN | 3.0 | NaN | NaN | 28 |
| **1** | 2014-06-01 01:00:00 | 0.2 | 0.2 | 0.1 | 0.11 | 3.0 | 17.0 | 68.0 | 10.0 | 5.0 | 5.0 | 1.36 | 1.3 | 28 |
| **2** | 2014-06-01 01:00:00 | 0.3 | NaN | 0.1 | NaN | 2.0 | 6.0 | NaN | NaN | NaN | NaN | NaN | 1.1 | 28 |
| **3** | 2014-06-01 01:00:00 | NaN | 0.2 | NaN | NaN | 1.0 | 6.0 | 79.0 | NaN | NaN | NaN | NaN | NaN | 28 |
| **4** | 2014-06-01 01:00:00 | NaN | NaN | NaN | NaN | 1.0 | 6.0 | 75.0 | NaN | NaN | 4.0 | NaN | NaN | 28 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **210019** | 2014-09-01 00:00:00 | NaN | 0.5 | NaN | NaN | 20.0 | 84.0 | 29.0 | NaN | NaN | NaN | NaN | NaN | 28 |
| **210020** | 2014-09-01 00:00:00 | NaN | 0.3 | NaN | NaN | 1.0 | 22.0 | NaN | 15.0 | NaN | 6.0 | NaN | NaN | 28 |
| **210021** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 1.0 | 13.0 | 70.0 | NaN | NaN | NaN | NaN | NaN | 28 |
| **210022** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 3.0 | 38.0 | 42.0 | NaN | NaN | NaN | NaN | NaN | 28 |
| **210023** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 1.0 | 26.0 | 65.0 | 11.0 | NaN | NaN | NaN | NaN | 28 |

210024 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** | 2014-06-01 01:00:00 | NaN | 0.2 | NaN | NaN | 3.0 | 10.0 | NaN | NaN | NaN | 3.0 | NaN | NaN | 28079004 |
| **1** | 2014-06-01 01:00:00 | 0.2 | 0.2 | 0.1 | 0.11 | 3.0 | 17.0 | 68.0 | 10.0 | 5.0 | 5.0 | 1.36 | 1.3 | 28079008 |
| **2** | 2014-06-01 01:00:00 | 0.3 | NaN | 0.1 | NaN | 2.0 | 6.0 | NaN | NaN | NaN | NaN | NaN | 1.1 | 28079011 |
| **3** | 2014-06-01 01:00:00 | NaN | 0.2 | NaN | NaN | 1.0 | 6.0 | 79.0 | NaN | NaN | NaN | NaN | NaN | 28079016 |
| **4** | 2014-06-01 01:00:00 | NaN | NaN | NaN | NaN | 1.0 | 6.0 | 75.0 | NaN | NaN | 4.0 | NaN | NaN | 28079017 |
| **5** | 2014-06-01 01:00:00 | 0.1 | 0.4 | 0.1 | NaN | 1.0 | 10.0 | 83.0 | 7.0 | NaN | 2.0 | NaN | 0.2 | 28079018 |
| **6** | 2014-06-01 01:00:00 | 0.1 | 0.2 | 0.1 | 0.23 | 1.0 | 5.0 | 80.0 | 4.0 | 3.0 | 2.0 | 1.21 | 0.1 | 28079024 |
| **7** | 2014-06-01 01:00:00 | NaN | NaN | NaN | NaN | 1.0 | 1.0 | 86.0 | NaN | NaN | NaN | NaN | NaN | 28079027 |
| **8** | 2014-06-01 01:00:00 | NaN | 0.3 | NaN | NaN | 5.0 | 22.0 | 68.0 | NaN | NaN | 4.0 | NaN | NaN | 28079035 |
| **9** | 2014-06-01 01:00:00 | NaN | 0.2 | NaN | NaN | 1.0 | 4.0 | NaN | 14.0 | 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 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **210004** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 1.0 | 39.0 | 42.0 | NaN | NaN | 7.0 | NaN | NaN | 28 |
| **210005** | 2014-09-01 00:00:00 | 0.2 | 0.3 | 0.1 | NaN | 3.0 | 38.0 | 61.0 | 20.0 | NaN | 3.0 | NaN | 1.1 | 28 |
| **210006** | 2014-09-01 00:00:00 | 0.2 | 0.2 | 0.1 | 0.23 | 1.0 | 30.0 | 69.0 | 18.0 | 13.0 | 3.0 | 1.30 | 0.1 | 28 |
| **210007** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 2.0 | 28.0 | 66.0 | NaN | NaN | NaN | NaN | NaN | 28 |
| **210008** | 2014-09-01 00:00:00 | NaN | 0.3 | NaN | NaN | 5.0 | 45.0 | 59.0 | NaN | NaN | 5.0 | NaN | NaN | 28 |
| **210009** | 2014-09-01 00:00:00 | NaN | 0.2 | NaN | NaN | 2.0 | 30.0 | NaN | 34.0 | NaN | 2.0 | NaN | NaN | 28 |
| **210010** | 2014-09-01 00:00:00 | 0.4 | NaN | 0.6 | NaN | 4.0 | 33.0 | NaN | 15.0 | 13.0 | 2.0 | NaN | 3.0 | 28 |
| **210011** | 2014-09-01 00:00:00 | NaN | 0.3 | NaN | NaN | 1.0 | 35.0 | 60.0 | NaN | NaN | NaN | NaN | NaN | 28 |
| **210012** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 3.0 | 44.0 | NaN | 23.0 | NaN | 3.0 | NaN | NaN | 28 |
| **210013** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 3.0 | 41.0 | NaN | 23.0 | 14.0 | NaN | NaN | NaN | 28 |
| **210014** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 1.0 | 24.0 | NaN | 17.0 | 10.0 | NaN | NaN | NaN | 28 |
| **210015** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 7.0 | 29.0 | 49.0 | NaN | NaN | NaN | NaN | NaN | 28 |
| **210016** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 1.0 | 26.0 | NaN | 19.0 | 11.0 | NaN | NaN | NaN | 28 |
| **210017** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 1.0 | 23.0 | 59.0 | NaN | NaN | NaN | NaN | NaN | 28 |
| **210018** | 2014-09-01 00:00:00 | NaN | NaN | NaN | 0.38 | 2.0 | 34.0 | NaN | 14.0 | NaN | NaN | 1.19 | NaN | 28 |
| **210019** | 2014-09-01 00:00:00 | NaN | 0.5 | NaN | NaN | 20.0 | 84.0 | 29.0 | NaN | NaN | NaN | NaN | NaN | 28 |
| **210020** | 2014-09-01 00:00:00 | NaN | 0.3 | NaN | NaN | 1.0 | 22.0 | NaN | 15.0 | NaN | 6.0 | NaN | NaN | 28 |

| | date | BEN | CO | EBE | NMHC | NO | NO_2 | O_3 | PM10 | PM25 | SO_2 | TCH | TOL | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **210021** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 1.0 | 13.0 | 70.0 | NaN | NaN | NaN | NaN | NaN | 28 |
| **210022** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 3.0 | 38.0 | 42.0 | NaN | NaN | NaN | NaN | NaN | 28 |
| **210023** | 2014-09-01 00:00:00 | NaN | NaN | NaN | NaN | 1.0 | 26.0 | 65.0 | 11.0 | NaN | NaN | NaN | NaN | 28 |

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

Out[5]:

| | BEN | CO | EBE | NMHC | NO | NO_2 | |
|---|---|---|---|---|---|---|---|
| **count** | 46703.000000 | 87023.000000 | 46722.000000 | 25021.000000 | 209154.000000 | 209154.000000 | 1 |
| **mean** | 0.682288 | 0.368252 | 0.470755 | 0.275535 | 19.973369 | 35.053367 | |
| **std** | 0.908237 | 0.244409 | 0.810957 | 0.153442 | 44.842427 | 28.395328 | |
| **min** | 0.100000 | 0.100000 | 0.100000 | 0.040000 | 1.000000 | 1.000000 | |
| **25%** | 0.100000 | 0.200000 | 0.100000 | 0.180000 | 2.000000 | 14.000000 | |
| **50%** | 0.300000 | 0.300000 | 0.200000 | 0.240000 | 5.000000 | 28.000000 | |
| **75%** | 0.800000 | 0.400000 | 0.500000 | 0.320000 | 17.000000 | 49.000000 | |
| **max** | 17.799999 | 4.400000 | 16.200001 | 1.590000 | 925.000000 | 416.000000 | |

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

Out[6]:
```
(210024, 14)
```

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

Out[7]:
```
2940336
```

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 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **210019** | False | True | False | True | True | False | False | False | True | True | True | True | True |
| **210020** | False | True | False | True | True | False | False | True | False | True | False | True | True |
| **210021** | False | True | True | True | True | False | False | False | True | True | True | True | True |
| **210022** | False | True | True | True | True | False | False | False | True | True | True | True | True |
| **210023** | False | True | True | True | True | False | False | False | False | True | True | True | True |

210024 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** | 2014-06-01 01:00:00 | 0.2 | 0.2 | 0.1 | 0.11 | 3.0 | 17.0 | 68.0 | 10.0 | 5.0 | 5.0 | 1.36 | 1.3 | 280 |
| **6** | 2014-06-01 01:00:00 | 0.1 | 0.2 | 0.1 | 0.23 | 1.0 | 5.0 | 80.0 | 4.0 | 3.0 | 2.0 | 1.21 | 0.1 | 280 |
| **25** | 2014-06-01 02:00:00 | 0.2 | 0.2 | 0.1 | 0.11 | 4.0 | 21.0 | 63.0 | 9.0 | 6.0 | 5.0 | 1.36 | 0.8 | 280 |
| **30** | 2014-06-01 02:00:00 | 0.2 | 0.2 | 0.1 | 0.23 | 1.0 | 4.0 | 88.0 | 7.0 | 5.0 | 2.0 | 1.21 | 0.1 | 280 |
| **49** | 2014-06-01 03:00:00 | 0.1 | 0.2 | 0.1 | 0.11 | 4.0 | 18.0 | 66.0 | 9.0 | 7.0 | 6.0 | 1.36 | 0.9 | 280 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **209958** | 2014-08-31 22:00:00 | 0.2 | 0.2 | 0.1 | 0.22 | 1.0 | 28.0 | 96.0 | 61.0 | 15.0 | 3.0 | 1.28 | 0.1 | 280 |
| **209977** | 2014-08-31 23:00:00 | 1.1 | 0.7 | 0.7 | 0.19 | 36.0 | 118.0 | 23.0 | 60.0 | 25.0 | 9.0 | 1.27 | 6.5 | 280 |
| **209982** | 2014-08-31 23:00:00 | 0.2 | 0.2 | 0.1 | 0.21 | 1.0 | 17.0 | 90.0 | 28.0 | 14.0 | 3.0 | 1.27 | 0.2 | 280 |
| **210001** | 2014-09-01 00:00:00 | 0.6 | 0.4 | 0.4 | 0.12 | 6.0 | 63.0 | 41.0 | 26.0 | 15.0 | 8.0 | 1.19 | 4.1 | 280 |
| **210006** | 2014-09-01 00:00:00 | 0.2 | 0.2 | 0.1 | 0.23 | 1.0 | 30.0 | 69.0 | 18.0 | 13.0 | 3.0 | 1.30 | 0.1 | 280 |

13946 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']]
```

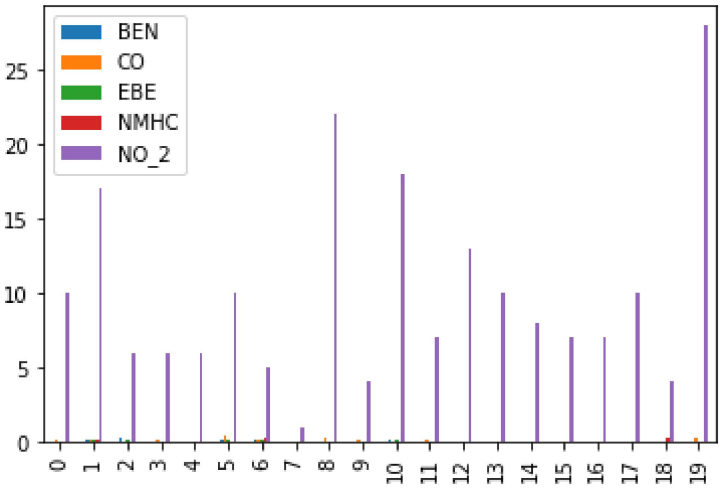
In [12]:
```
dd=sd.head(20)
dd
```

Out[12]:

| | BEN | CO | EBE | NMHC | NO_2 |
|---|---|---|---|---|---|
| **0** | NaN | 0.2 | NaN | NaN | 10.0 |
| **1** | 0.2 | 0.2 | 0.1 | 0.11 | 17.0 |
| **2** | 0.3 | NaN | 0.1 | NaN | 6.0 |
| **3** | NaN | 0.2 | NaN | NaN | 6.0 |
| **4** | NaN | NaN | NaN | NaN | 6.0 |
| **5** | 0.1 | 0.4 | 0.1 | NaN | 10.0 |
| **6** | 0.1 | 0.2 | 0.1 | 0.23 | 5.0 |
| **7** | NaN | NaN | NaN | NaN | 1.0 |
| **8** | NaN | 0.3 | NaN | NaN | 22.0 |
| **9** | NaN | 0.2 | NaN | NaN | 4.0 |
| **10** | 0.1 | NaN | 0.1 | NaN | 18.0 |
| **11** | NaN | 0.2 | NaN | NaN | 7.0 |
| **12** | NaN | NaN | NaN | NaN | 13.0 |
| **13** | NaN | NaN | NaN | NaN | 10.0 |
| **14** | NaN | NaN | NaN | NaN | 8.0 |
| **15** | NaN | NaN | NaN | NaN | 7.0 |
| **16** | NaN | NaN | NaN | NaN | 7.0 |
| **17** | NaN | NaN | NaN | NaN | 10.0 |
| **18** | NaN | NaN | NaN | 0.23 | 4.0 |
| **19** | NaN | 0.3 | NaN | NaN | 28.0 |

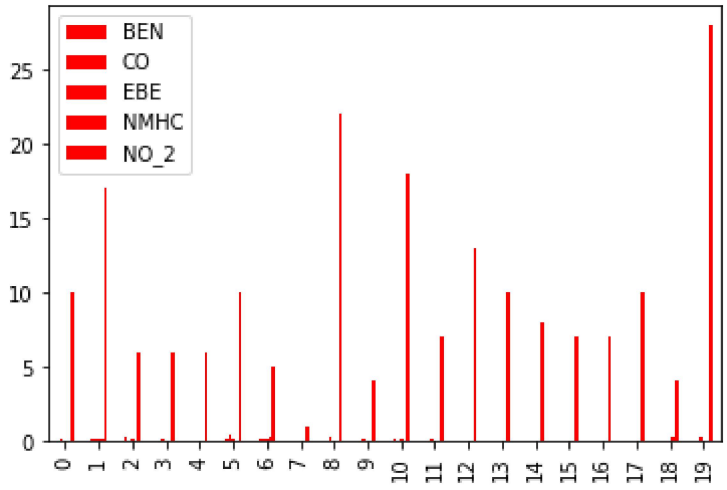
In [13]:
```
dd.plot.bar()
```

Out[13]: <AxesSubplot:>



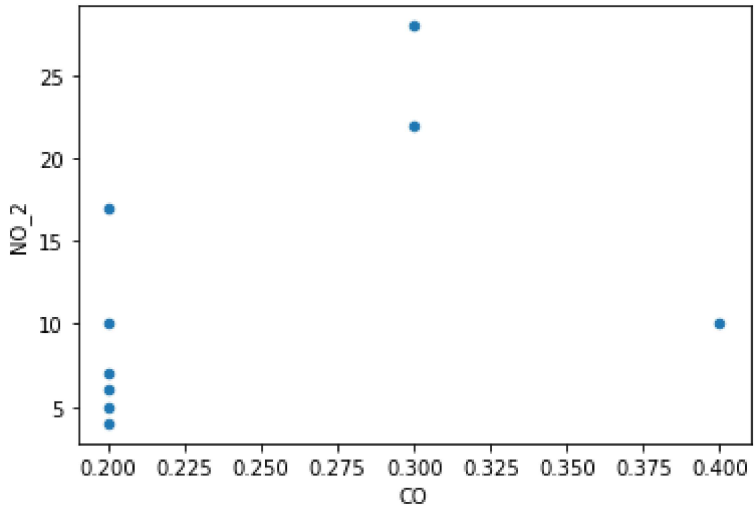
In [14]:
```
dd.plot.bar(color='r')
```

Out[14]: <AxesSubplot:>



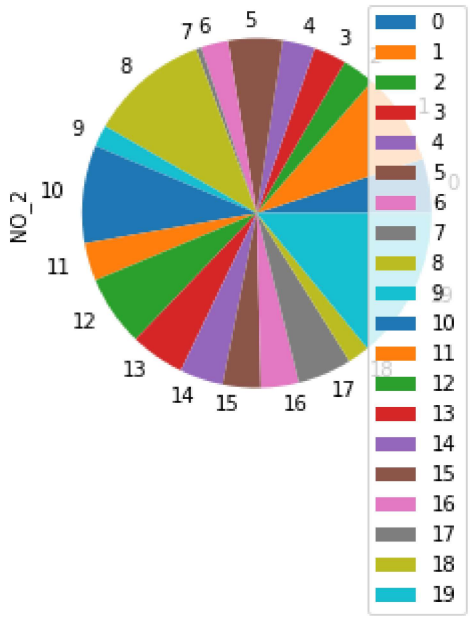

In [15]:
```
dd.plot.scatter(x='CO',y='NO_2')
```

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



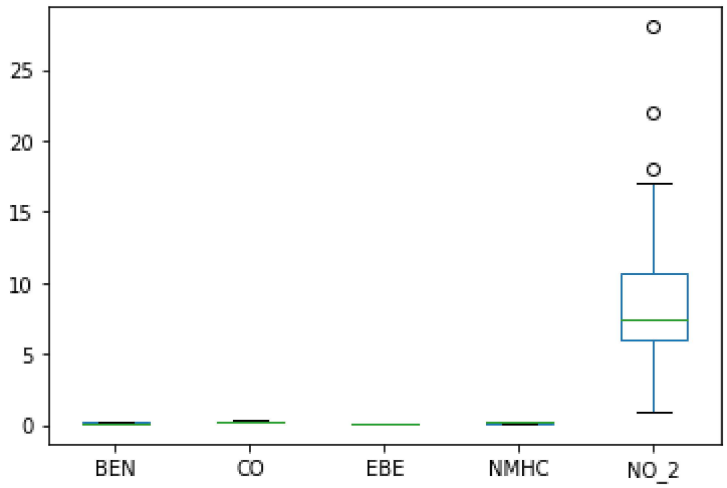
In [16]:
```
dd.plot.pie(y='NO_2')
```

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



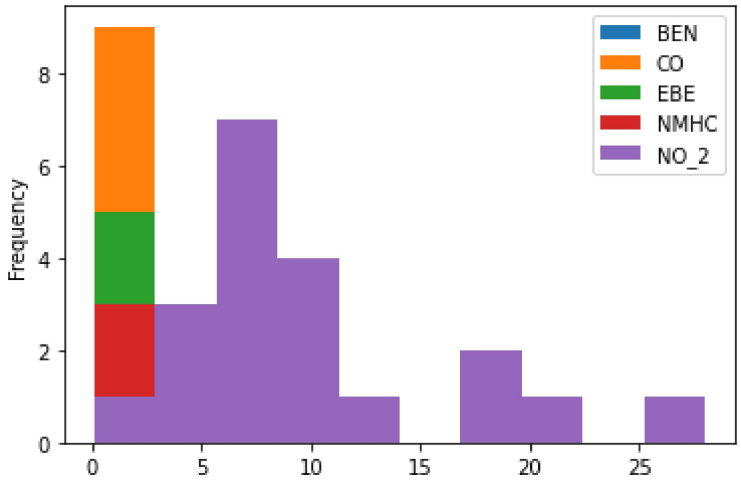

In [17]:
```
dd.plot.box()
```

Out[17]: <AxesSubplot:>



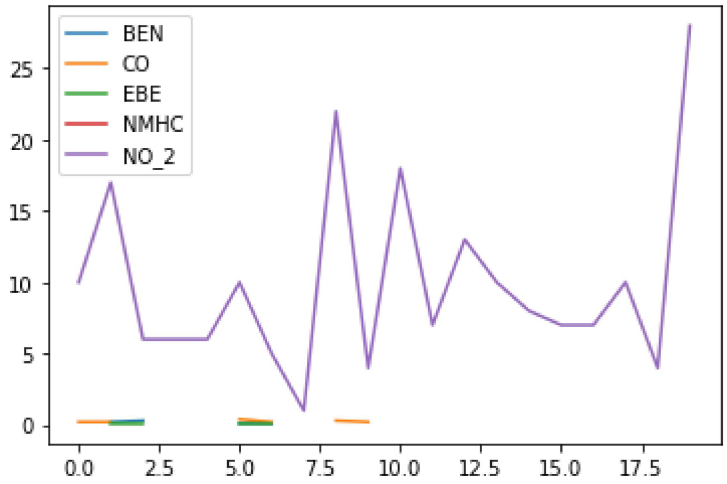
In [18]:
```
dd.plot.hist()
```

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



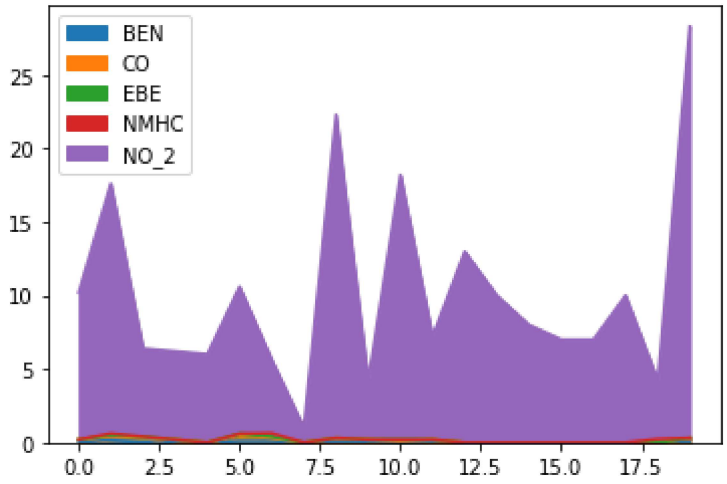

In [19]:
```
dd.plot.line()
```

Out[19]: <AxesSubplot:>



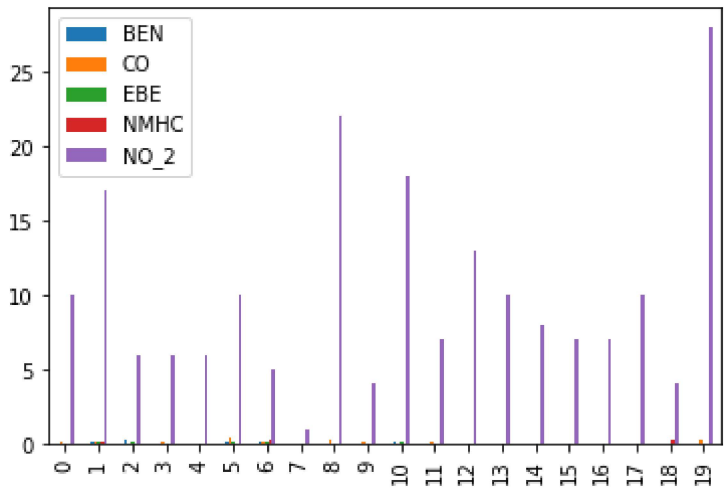
In [20]:
```
dd.plot.area()
```

Out[20]: <AxesSubplot:>



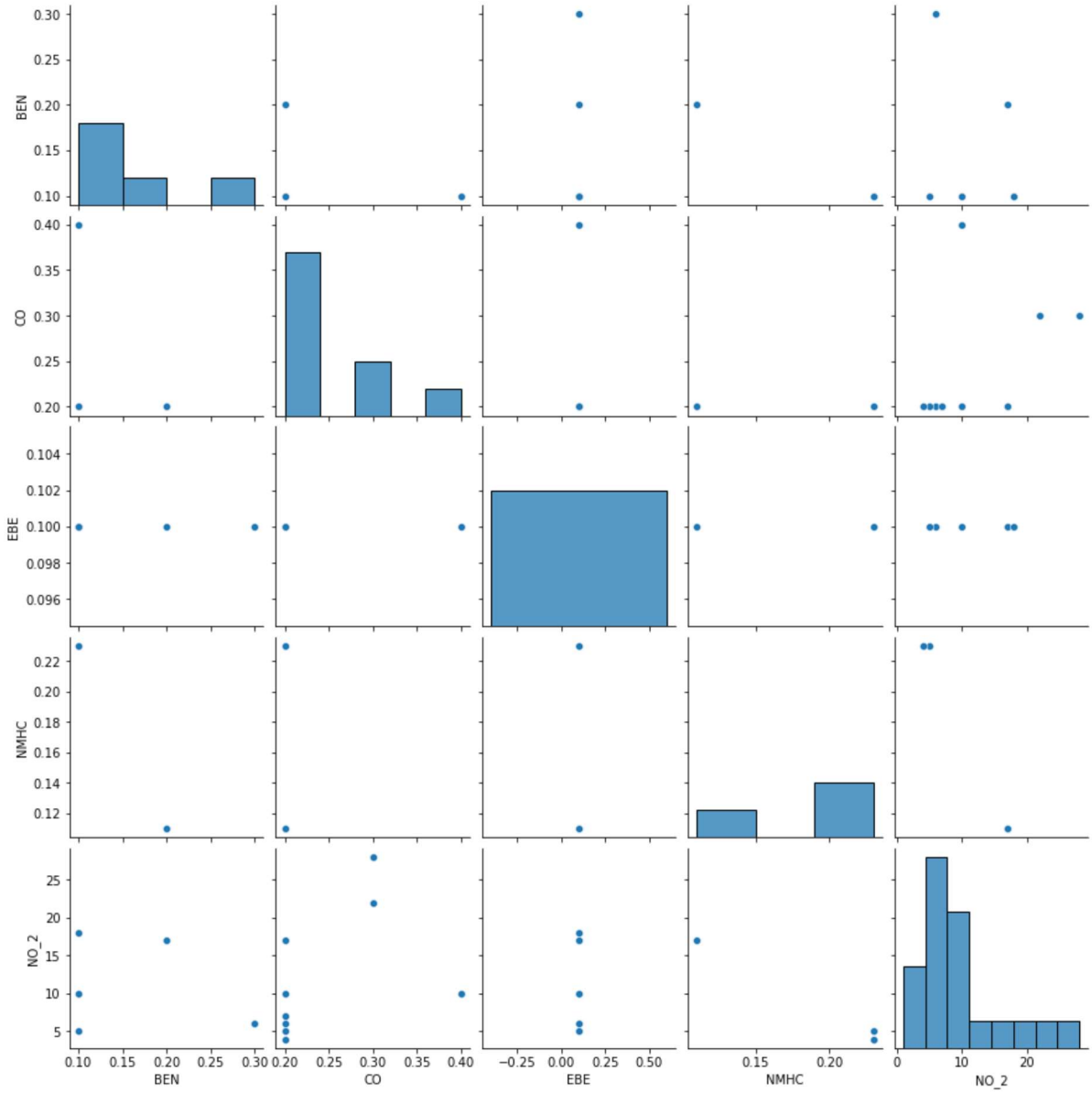

In [21]:
```
dd.plot.bar()
```

Out[21]: <AxesSubplot:>



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

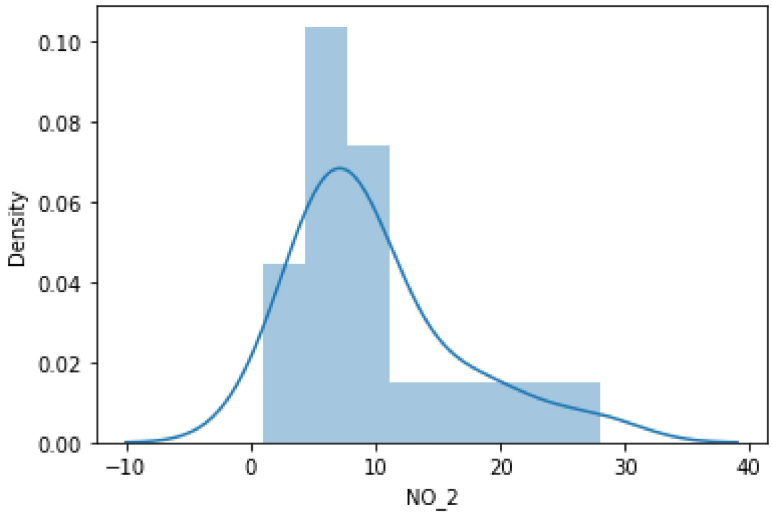
Out[22]:
```
<seaborn.axisgrid.PairGrid at 0x25145ad5c10>
```



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'>



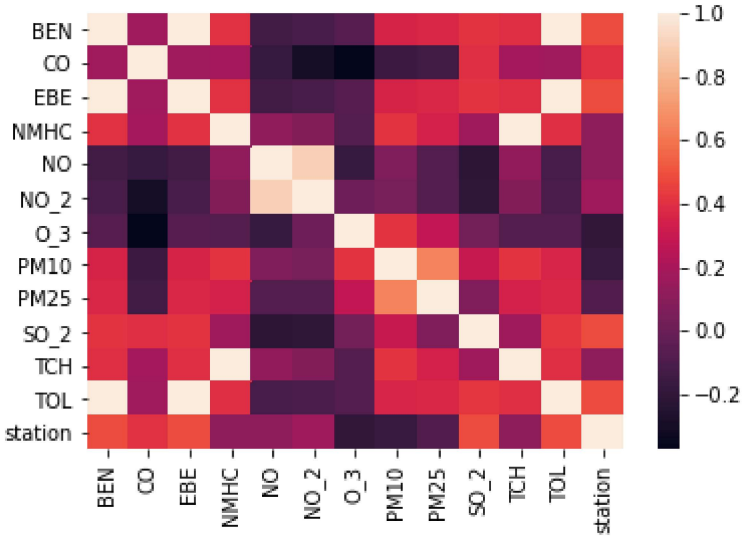

In [24]:
```
ds=data.fillna(20)
```

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_)
```

```
28079018.38625455
```

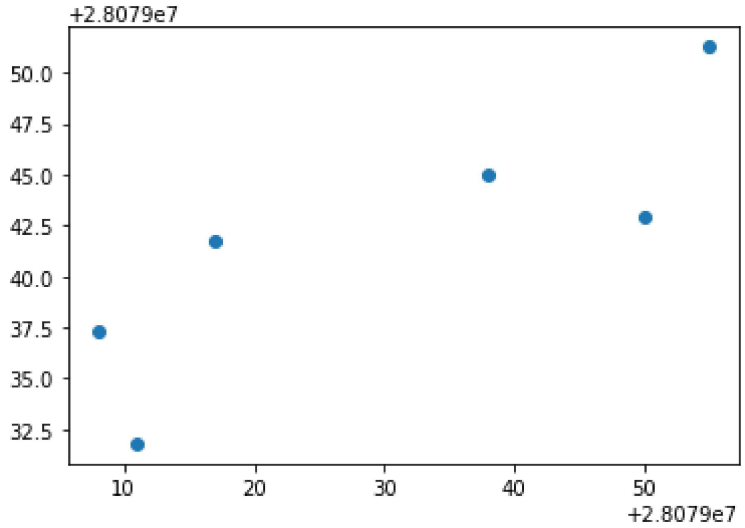
In [32]:
```
coeff= pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[32]:

| | Co-efficient |
|---|---|
| **BEN** | 0.251961 |
| **CO** | 0.926642 |
| **EBE** | 0.251961 |
| **NMHC** | -0.591779 |
| **NO_2** | 1.102827 |

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

Out[33]:
```
<matplotlib.collections.PathCollection at 0x2514864d940>
```



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

```
0.039251251590954994
```

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

Out[35]:
```
0.039251251590954994
```

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

Out[36]:
```
0.5400680566125595
```

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]:
```
0.0477481459974376
```

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

Out[40]:
```
0.5398643864554507
```

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]:
```
-0.130550472867212
```

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

Out[43]:
```
0.4981842944765028
```

# 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_)
```

```
[ 0.23612612  0.91083087  0.22607102 -0.50726305  1.06900651]
```

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

```
28079018.00957325
```

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

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

```
0.04019549855706883
```

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]:
```
0.002024004326269531
```

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')
```

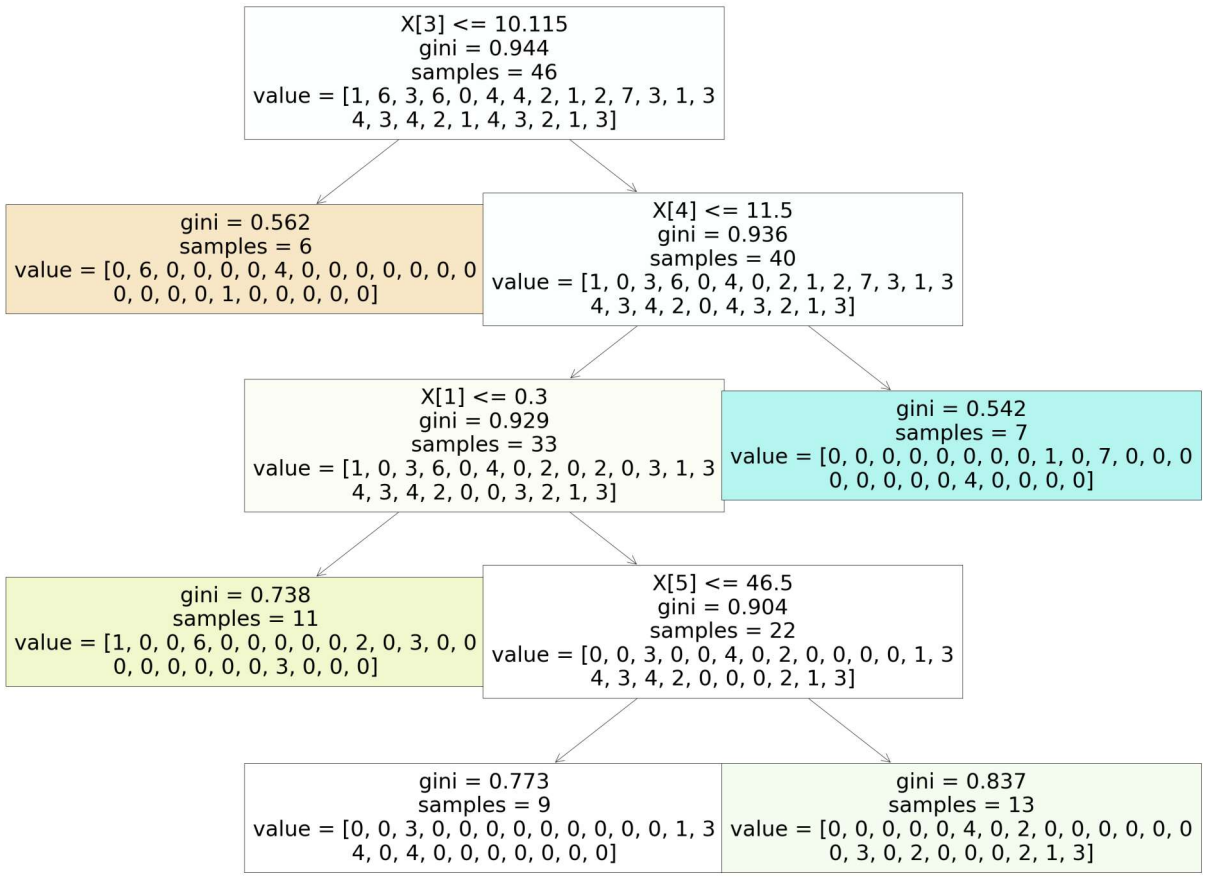
In [69]:
```
grid_search.best_score_
```

Out[69]:
```
0.5428571428571428
```

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(1116.0, 1956.96, 'X[3] <= 10.115\ngini = 0.944\nsamples = 46\nvalue =
[1, 6, 3, 6, 0, 4, 4, 2, 1, 2, 7, 3, 1, 3\n4, 3, 4, 2, 1, 4, 3, 2, 1, 3]'),
 Text(558.0, 1522.0800000000002, 'gini = 0.562\nsamples = 6\nvalue = [0, 6,
0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 1, 0, 0, 0, 0, 0]'),
 Text(1674.0, 1522.0800000000002, 'X[4] <= 11.5\ngini = 0.936\nsamples = 40\n
value = [1, 0, 3, 6, 0, 4, 0, 2, 1, 2, 7, 3, 1, 3\n4, 3, 4, 2, 0, 4, 3, 2, 1,
3]'),
 Text(1116.0, 1087.2, 'X[1] <= 0.3\ngini = 0.929\nsamples = 33\nvalue = [1,
0, 3, 6, 0, 4, 0, 2, 0, 2, 0, 3, 1, 3\n4, 3, 4, 2, 0, 0, 3, 2, 1, 3]'),
 Text(558.0, 652.3200000000002, 'gini = 0.738\nsamples = 11\nvalue = [1, 0,
0, 6, 0, 0, 0, 0, 0, 2, 0, 3, 0, 0\n0, 0, 0, 0, 0, 0, 3, 0, 0, 0]'),
 Text(1674.0, 652.3200000000002, 'X[5] <= 46.5\ngini = 0.904\nsamples = 22\nv
alue = [0, 0, 3, 0, 0, 4, 0, 2, 0, 0, 0, 0, 1, 3\n4, 3, 4, 2, 0, 0, 0, 2, 1,
3]'),
 Text(1116.0, 217.44000000000005, 'gini = 0.773\nsamples = 9\nvalue = [0, 0,
3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3\n4, 0, 4, 0, 0, 0, 0, 0, 0, 0]'),
 Text(2232.0, 217.44000000000005, 'gini = 0.837\nsamples = 13\nvalue = [0, 0,
0, 0, 0, 4, 0, 2, 0, 0, 0, 0, 0, 0\n0, 3, 0, 2, 0, 0, 0, 2, 1, 3]'),
 Text(2232.0, 1087.2, 'gini = 0.542\nsamples = 7\nvalue = [0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 7, 0, 0, 0\n0, 0, 0, 0, 0, 4, 0, 0, 0, 0]')]
```



# Conclusion : LogisticRegression() [28079056] HIGH RANGE

In [ ]: