In [1]:

```python
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```
data=pd.read_csv(r"C:\Users\user\Desktop\DINESH\C10_air\stations.csv")
data
```

Out[3]:

| | id | name | address | lon | lat | elevation |
|---|---|---|---|---|---|---|
| **0** | 28079004 | Pza. de España | Plaza de España | -3.712247 | 40.423853 | 635 |
| **1** | 28079008 | Escuelas Aguirre | Entre C/ Alcalá y C/ O' Donell | -3.682319 | 40.421564 | 670 |
| **2** | 28079011 | Avda. Ramón y Cajal | Avda. Ramón y Cajal esq. C/ Príncipe de Vergara | -3.677356 | 40.451475 | 708 |
| **3** | 28079016 | Arturo Soria | C/ Arturo Soria esq. C/ Vizconde de los Asilos | -3.639233 | 40.440047 | 693 |
| **4** | 28079017 | Villaverde | C/. Juan Peñalver | -3.713322 | 40.347139 | 604 |
| **5** | 28079018 | Farolillo | Calle Farolillo - C/Ervigio | -3.731853 | 40.394781 | 630 |
| **6** | 28079024 | Casa de Campo | Casa de Campo (Terminal del Teleférico) | -3.747347 | 40.419356 | 642 |
| **7** | 28079027 | Barajas Pueblo | C/. Júpiter, 21 (Barajas) | -3.580031 | 40.476928 | 621 |
| **8** | 28079035 | Pza. del Carmen | Plaza del Carmen esq. Tres Cruces. | -3.703172 | 40.419208 | 659 |
| **9** | 28079036 | Moratalaz | Avd. Moratalaz esq. Camino de los Vinateros | -3.645306 | 40.407947 | 685 |
| **10** | 28079038 | Cuatro Caminos | Avda. Pablo Iglesias esq. C/ Marqués de Lema | -3.707128 | 40.445544 | 698 |
| **11** | 28079039 | Barrio del Pilar | Avd. Betanzos esq. C/ Monforte de Lemos | -3.711542 | 40.478228 | 674 |
| **12** | 28079040 | Vallecas | C/ Arroyo del Olivar esq. C/ Río Grande. | -3.651522 | 40.388153 | 677 |
| **13** | 28079047 | Mendez Alvaro | C/ Juan de Mariana / Pza. Amanecer Mendez Alvaro | -3.686825 | 40.398114 | 599 |
| **14** | 28079048 | Castellana | C/ Jose Gutierrez Abascal | -3.690367 | 40.439897 | 676 |
| **15** | 28079049 | Parque del Retiro | Paseo Venezuela- Casa de Vacas | -3.682583 | 40.414444 | 662 |
| **16** | 28079050 | Plaza Castilla | Plaza Castilla (Canal) | -3.688769 | 40.465572 | 728 |
| **17** | 28079054 | Ensanche de Vallecas | Avda La Gavia / Avda. Las Suertes | -3.612117 | 40.372933 | 627 |
| **18** | 28079055 | Urb. Embajada | C/ Riaño (Barajas) | -3.580747 | 40.462531 | 618 |
| **19** | 28079056 | Pza. Fernández Ladreda | Pza. Fernández Ladreda - Avda. Oporto | -3.718728 | 40.384964 | 604 |
| **20** | 28079057 | Sanchinarro | C/ Princesa de Eboli esq C/ Maria Tudor | -3.660503 | 40.494208 | 700 |
| **21** | 28079058 | El Pardo | Avda. La Guardia | -3.774611 | 40.518058 | 615 |
| **22** | 28079059 | Juan Carlos I | Parque Juan Carlos I (frente oficinas mantenim... | -3.609072 | 40.465250 | 660 |
| **23** | 28079060 | Tres Olivos | Plaza Tres Olivos | -3.689761 | 40.500589 | 715 |

In [4]:

```
data.head(10)
```

Out[4]:

|   | id | name | address | lon | lat | elevation |
|---|---|---|---|---|---|---|
| **0** | 28079004 | Pza. de España | Plaza de España | -3.712247 | 40.423853 | 635 |
| **1** | 28079008 | Escuelas Aguirre | Entre C/ Alcalá y C/ O' Donell | -3.682319 | 40.421564 | 670 |
| **2** | 28079011 | Avda. Ramón y Cajal | Avda. Ramón y Cajal esq. C/ Príncipe de Vergara | -3.677356 | 40.451475 | 708 |
| **3** | 28079016 | Arturo Soria | C/ Arturo Soria esq. C/ Vizconde de los Asilos | -3.639233 | 40.440047 | 693 |
| **4** | 28079017 | Villaverde | C/. Juan Peñalver | -3.713322 | 40.347139 | 604 |
| **5** | 28079018 | Farolillo | Calle Farolillo - C/Ervigio | -3.731853 | 40.394781 | 630 |
| **6** | 28079024 | Casa de Campo | Casa de Campo (Terminal del Teleférico) | -3.747347 | 40.419356 | 642 |
| **7** | 28079027 | Barajas Pueblo | C/. Júpiter, 21 (Barajas) | -3.580031 | 40.476928 | 621 |
| **8** | 28079035 | Pza. del Carmen | Plaza del Carmen esq. Tres Cruces. | -3.703172 | 40.419208 | 659 |
| **9** | 28079036 | Moratalaz | Avd. Moratalaz esq. Camino de los Vinateros | -3.645306 | 40.407947 | 685 |

In [5]:
```
data.tail(20)
```

Out[5]:

| | id | name | address | lon | lat | elevation |
|---|---|---|---|---|---|---|
| **4** | 28079017 | Villaverde | C/. Juan Peñalver | -3.713322 | 40.347139 | 604 |
| **5** | 28079018 | Farolillo | Calle Farolillo - C/Ervigio | -3.731853 | 40.394781 | 630 |
| **6** | 28079024 | Casa de Campo | Casa de Campo (Terminal del Teleférico) | -3.747347 | 40.419356 | 642 |
| **7** | 28079027 | Barajas Pueblo | C/. Júpiter, 21 (Barajas) | -3.580031 | 40.476928 | 621 |
| **8** | 28079035 | Pza. del Carmen | Plaza del Carmen esq. Tres Cruces. | -3.703172 | 40.419208 | 659 |
| **9** | 28079036 | Moratalaz | Avd. Moratalaz esq. Camino de los Vinateros | -3.645306 | 40.407947 | 685 |
| **10** | 28079038 | Cuatro Caminos | Avda. Pablo Iglesias esq. C/ Marqués de Lema | -3.707128 | 40.445544 | 698 |
| **11** | 28079039 | Barrio del Pilar | Avd. Betanzos esq. C/ Monforte de Lemos | -3.711542 | 40.478228 | 674 |
| **12** | 28079040 | Vallecas | C/ Arroyo del Olivar esq. C/ Río Grande. | -3.651522 | 40.388153 | 677 |
| **13** | 28079047 | Mendez Alvaro | C/ Juan de Mariana / Pza. Amanecer Mendez Alvaro | -3.686825 | 40.398114 | 599 |
| **14** | 28079048 | Castellana | C/ Jose Gutierrez Abascal | -3.690367 | 40.439897 | 676 |
| **15** | 28079049 | Parque del Retiro | Paseo Venezuela- Casa de Vacas | -3.682583 | 40.414444 | 662 |
| **16** | 28079050 | Plaza Castilla | Plaza Castilla (Canal) | -3.688769 | 40.465572 | 728 |
| **17** | 28079054 | Ensanche de Vallecas | Avda La Gavia / Avda. Las Suertes | -3.612117 | 40.372933 | 627 |
| **18** | 28079055 | Urb. Embajada | C/ Riaño (Barajas) | -3.580747 | 40.462531 | 618 |
| **19** | 28079056 | Pza. Fernández Ladreda | Pza. Fernández Ladreda - Avda. Oporto | -3.718728 | 40.384964 | 604 |
| **20** | 28079057 | Sanchinarro | C/ Princesa de Eboli esq C/ Maria Tudor | -3.660503 | 40.494208 | 700 |
| **21** | 28079058 | El Pardo | Avda. La Guardia | -3.774611 | 40.518058 | 615 |
| **22** | 28079059 | Juan Carlos I | Parque Juan Carlos I (frente oficinas mantenim... | -3.609072 | 40.465250 | 660 |
| **23** | 28079060 | Tres Olivos | Plaza Tres Olivos | -3.689761 | 40.500589 | 715 |

In [6]:
```
data.describe()
```

Out[6]:

| | id | lon | lat | elevation |
|---|---|---|---|---|
| **count** | 2.400000e+01 | 24.000000 | 24.000000 | 24.000000 |
| **mean** | 2.807904e+07 | -3.679019 | 40.434616 | 658.333333 |
| **std** | 1.799094e+01 | 0.049324 | 0.043022 | 38.295949 |
| **min** | 2.807900e+07 | -3.774611 | 40.347139 | 599.000000 |
| **25%** | 2.807902e+07 | -3.711718 | 40.405489 | 625.500000 |
| **50%** | 2.807904e+07 | -3.687797 | 40.431875 | 661.000000 |
| **75%** | 2.807905e+07 | -3.649968 | 40.465331 | 687.000000 |
| **max** | 2.807906e+07 | -3.580031 | 40.518058 | 728.000000 |

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

Out[7]:
```
(24, 6)
```

In [8]:
```
np.size(data)
```

Out[8]:
```
144
```

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

Out[9]:

|  | id | name | address | lon | lat | elevation |
|---|---|---|---|---|---|---|
| **0** | False | False | False | False | False | False |
| **1** | False | False | False | False | False | False |
| **2** | False | False | False | False | False | False |
| **3** | False | False | False | False | False | False |
| **4** | False | False | False | False | False | False |
| **5** | False | False | False | False | False | False |
| **6** | False | False | False | False | False | False |
| **7** | False | False | False | False | False | False |
| **8** | False | False | False | False | False | False |
| **9** | False | False | False | False | False | False |
| **10** | False | False | False | False | False | False |
| **11** | False | False | False | False | False | False |
| **12** | False | False | False | False | False | False |
| **13** | False | False | False | False | False | False |
| **14** | False | False | False | False | False | False |
| **15** | False | False | False | False | False | False |
| **16** | False | False | False | False | False | False |
| **17** | False | False | False | False | False | False |
| **18** | False | False | False | False | False | False |
| **19** | False | False | False | False | False | False |
| **20** | False | False | False | False | False | False |
| **21** | False | False | False | False | False | False |
| **22** | False | False | False | False | False | False |
| **23** | False | False | False | False | False | False |

In [10]:

```
data.dropna()
```

Out[10]:

| | id | name | address | lon | lat | elevation |
|---|---|---|---|---|---|---|
| **0** | 28079004 | Pza. de España | Plaza de España | -3.712247 | 40.423853 | 635 |
| **1** | 28079008 | Escuelas Aguirre | Entre C/ Alcalá y C/ O' Donell | -3.682319 | 40.421564 | 670 |
| **2** | 28079011 | Avda. Ramón y Cajal | Avda. Ramón y Cajal esq. C/ Príncipe de Vergara | -3.677356 | 40.451475 | 708 |
| **3** | 28079016 | Arturo Soria | C/ Arturo Soria esq. C/ Vizconde de los Asilos | -3.639233 | 40.440047 | 693 |
| **4** | 28079017 | Villaverde | C/. Juan Peñalver | -3.713322 | 40.347139 | 604 |
| **5** | 28079018 | Farolillo | Calle Farolillo - C/Ervigio | -3.731853 | 40.394781 | 630 |
| **6** | 28079024 | Casa de Campo | Casa de Campo (Terminal del Teleférico) | -3.747347 | 40.419356 | 642 |
| **7** | 28079027 | Barajas Pueblo | C/. Júpiter, 21 (Barajas) | -3.580031 | 40.476928 | 621 |
| **8** | 28079035 | Pza. del Carmen | Plaza del Carmen esq. Tres Cruces. | -3.703172 | 40.419208 | 659 |
| **9** | 28079036 | Moratalaz | Avd. Moratalaz esq. Camino de los Vinateros | -3.645306 | 40.407947 | 685 |
| **10** | 28079038 | Cuatro Caminos | Avda. Pablo Iglesias esq. C/ Marqués de Lema | -3.707128 | 40.445544 | 698 |
| **11** | 28079039 | Barrio del Pilar | Avd. Betanzos esq. C/ Monforte de Lemos | -3.711542 | 40.478228 | 674 |
| **12** | 28079040 | Vallecas | C/ Arroyo del Olivar esq. C/ Río Grande. | -3.651522 | 40.388153 | 677 |
| **13** | 28079047 | Mendez Alvaro | C/ Juan de Mariana / Pza. Amanecer Mendez Alvaro | -3.686825 | 40.398114 | 599 |
| **14** | 28079048 | Castellana | C/ Jose Gutierrez Abascal | -3.690367 | 40.439897 | 676 |
| **15** | 28079049 | Parque del Retiro | Paseo Venezuela- Casa de Vacas | -3.682583 | 40.414444 | 662 |
| **16** | 28079050 | Plaza Castilla | Plaza Castilla (Canal) | -3.688769 | 40.465572 | 728 |
| **17** | 28079054 | Ensanche de Vallecas | Avda La Gavia / Avda. Las Suertes | -3.612117 | 40.372933 | 627 |
| **18** | 28079055 | Urb. Embajada | C/ Riaño (Barajas) | -3.580747 | 40.462531 | 618 |
| **19** | 28079056 | Pza. Fernández Ladreda | Pza. Fernández Ladreda - Avda. Oporto | -3.718728 | 40.384964 | 604 |
| **20** | 28079057 | Sanchinarro | C/ Princesa de Eboli esq C/ Maria Tudor | -3.660503 | 40.494208 | 700 |
| **21** | 28079058 | El Pardo | Avda. La Guardia | -3.774611 | 40.518058 | 615 |
| **22** | 28079059 | Juan Carlos I | Parque Juan Carlos I (frente oficinas mantenim... | -3.609072 | 40.465250 | 660 |
| **23** | 28079060 | Tres Olivos | Plaza Tres Olivos | -3.689761 | 40.500589 | 715 |

In [11]:

```
data.columns
```

Out[11]:

```
Index(['id', 'name', 'address', 'lon', 'lat', 'elevation'], dtype='object')
```

In [12]:

```
sd=data[['id','lon', 'lat', 'elevation']]
```

In [13]:

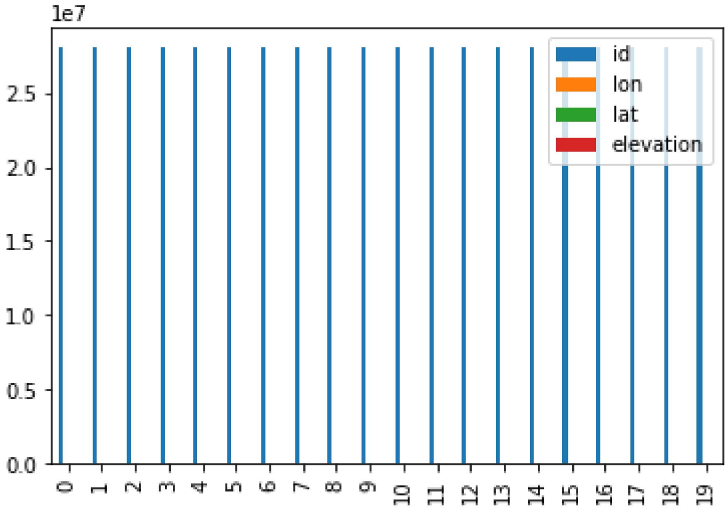
```
dd=sd.head(20)
dd
```

Out[13]:

| | id | lon | lat | elevation |
|---|---|---|---|---|
| **0** | 28079004 | -3.712247 | 40.423853 | 635 |
| **1** | 28079008 | -3.682319 | 40.421564 | 670 |
| **2** | 28079011 | -3.677356 | 40.451475 | 708 |
| **3** | 28079016 | -3.639233 | 40.440047 | 693 |
| **4** | 28079017 | -3.713322 | 40.347139 | 604 |
| **5** | 28079018 | -3.731853 | 40.394781 | 630 |
| **6** | 28079024 | -3.747347 | 40.419356 | 642 |
| **7** | 28079027 | -3.580031 | 40.476928 | 621 |
| **8** | 28079035 | -3.703172 | 40.419208 | 659 |
| **9** | 28079036 | -3.645306 | 40.407947 | 685 |
| **10** | 28079038 | -3.707128 | 40.445544 | 698 |
| **11** | 28079039 | -3.711542 | 40.478228 | 674 |
| **12** | 28079040 | -3.651522 | 40.388153 | 677 |
| **13** | 28079047 | -3.686825 | 40.398114 | 599 |
| **14** | 28079048 | -3.690367 | 40.439897 | 676 |
| **15** | 28079049 | -3.682583 | 40.414444 | 662 |
| **16** | 28079050 | -3.688769 | 40.465572 | 728 |
| **17** | 28079054 | -3.612117 | 40.372933 | 627 |
| **18** | 28079055 | -3.580747 | 40.462531 | 618 |
| **19** | 28079056 | -3.718728 | 40.384964 | 604 |

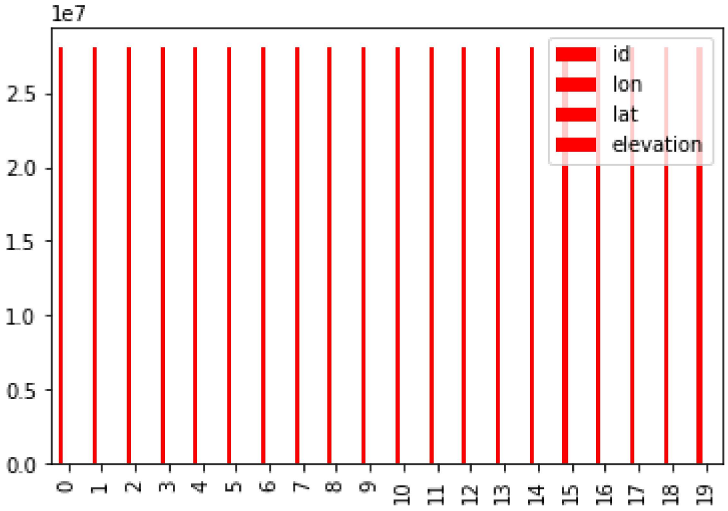
In [14]:
```
dd.plot.bar()
```

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



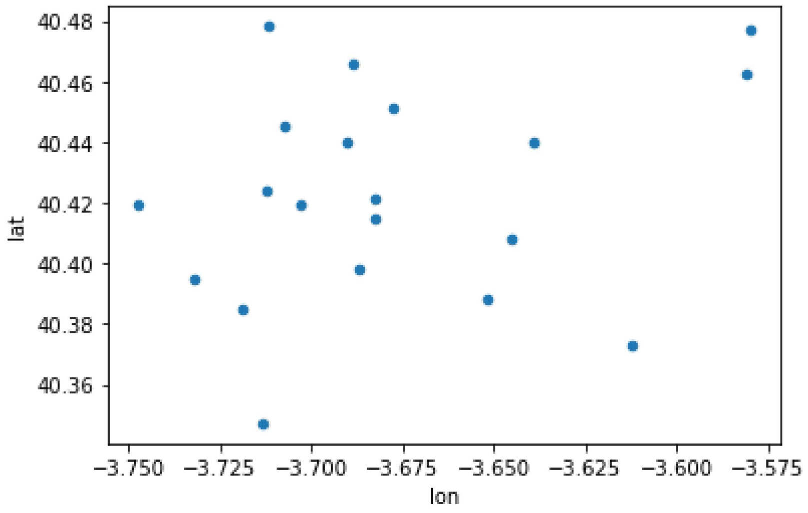
In [15]:
```
dd.plot.bar(color='r')
```

Out[15]:
```
<AxesSubplot:>
```



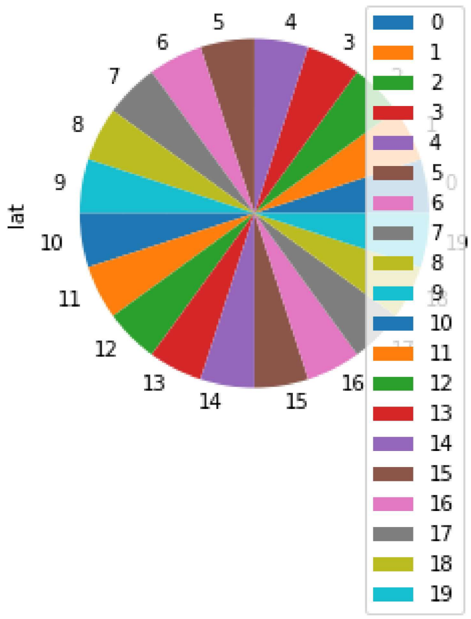
```
In [17]: dd.plot.scatter(x='lon',y='lat')
```

```
Out[17]: <AxesSubplot:xlabel='lon', ylabel='lat'>
```



```
In [18]: dd.plot.pie(y='lat')
```

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

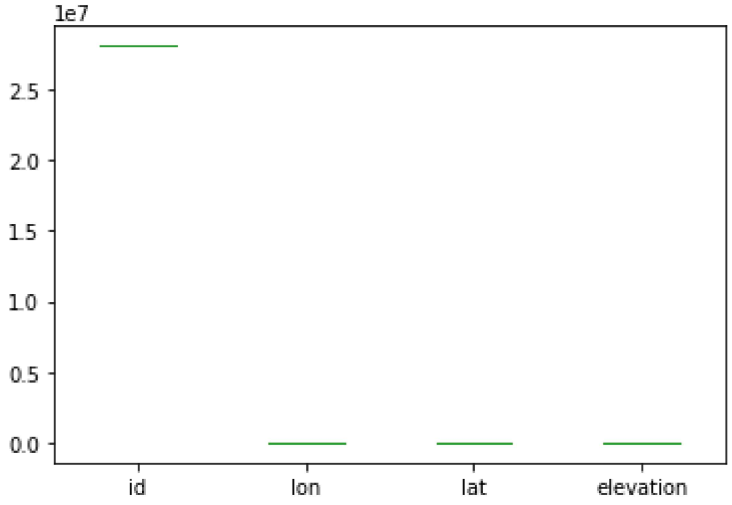


In [19]:

```
dd.plot.box()
```

Out[19]:

```
<AxesSubplot:>
```

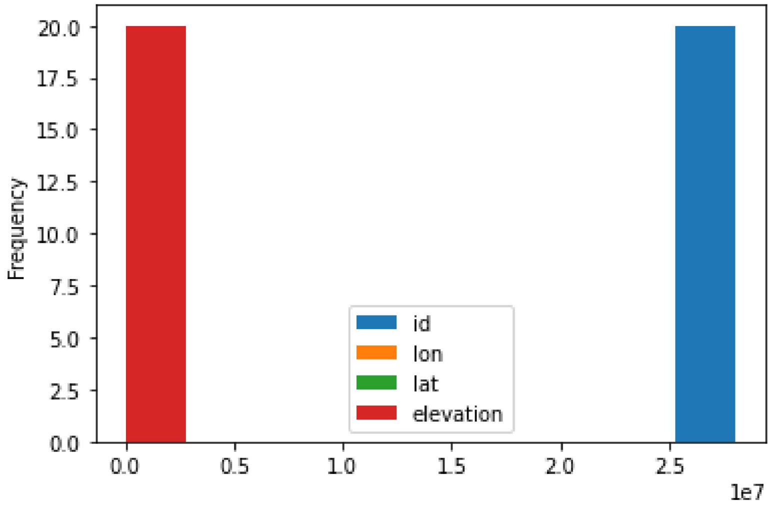


In [20]:

```
dd.plot.hist()
```

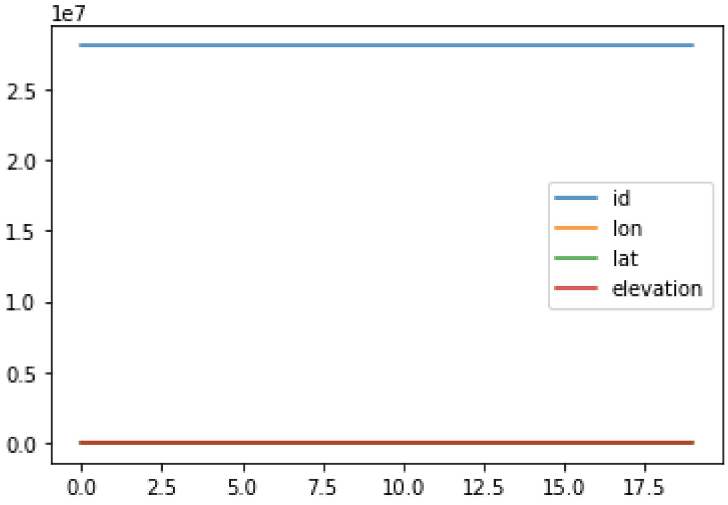
Out[20]:

```
<AxesSubplot:ylabel='Frequency'>
```



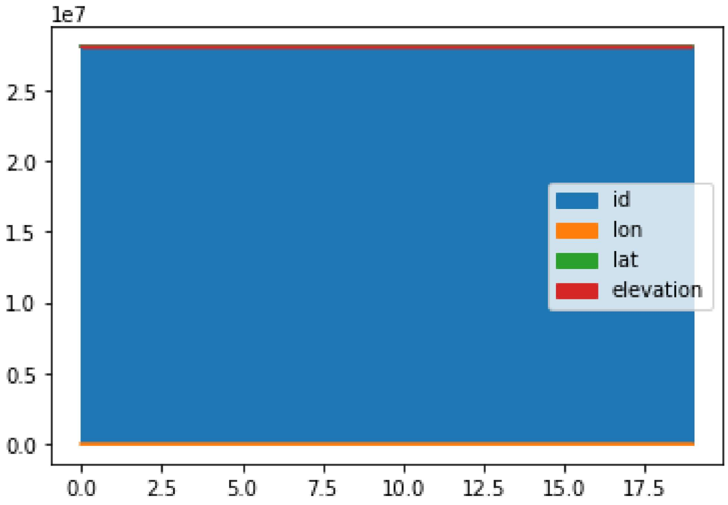
```
In [21]: dd.plot.line()
```

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



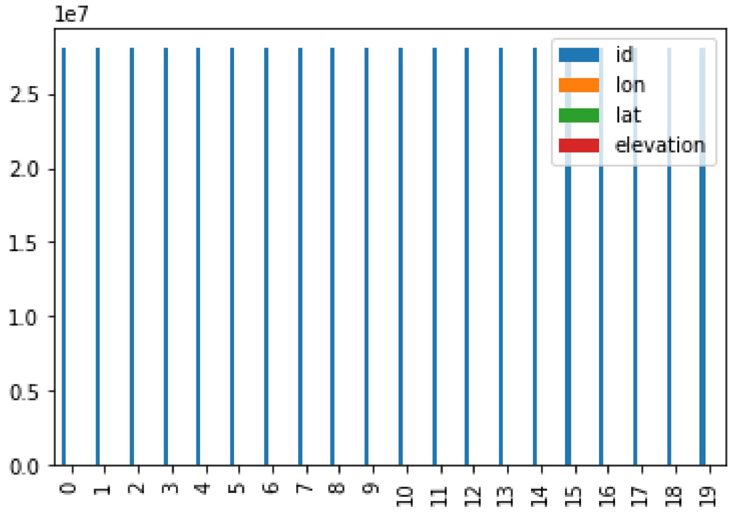
```
In [22]: dd.plot.area()
```

```
Out[22]: <AxesSubplot:>
```

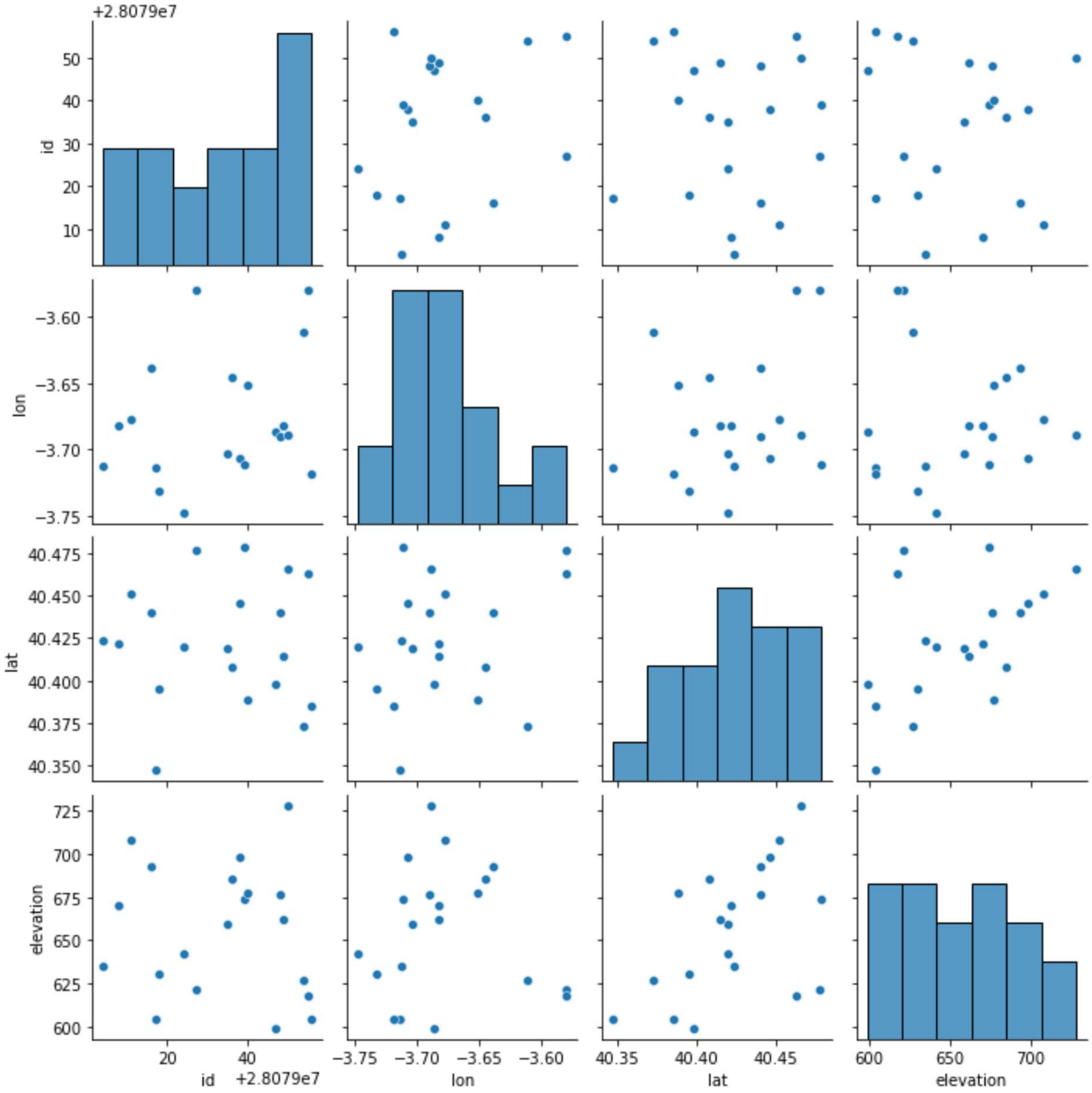


```
In [23]: dd.plot.bar()
```

```
Out[23]: <AxesSubplot:>
```



```
In [24]: sns.pairplot(dd)

Out[24]: <seaborn.axisgrid.PairGrid at 0x20f2d1597c0>
```



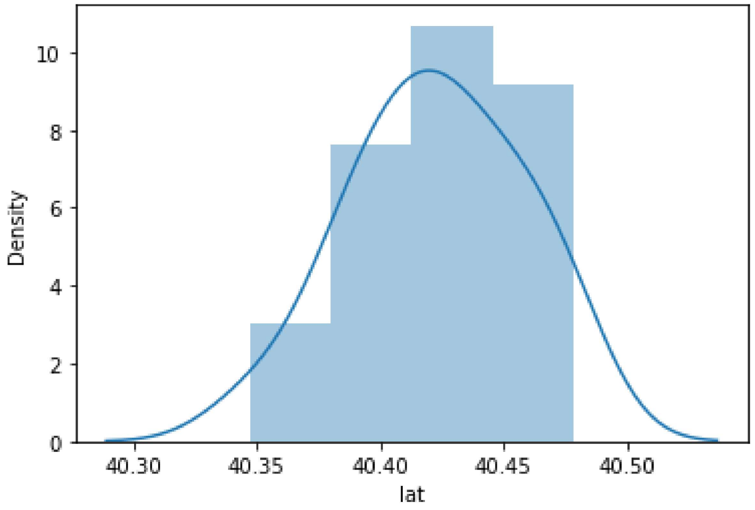
In [25]:

```
sns.distplot(dd['lat'])
```

```
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[25]:

```
<AxesSubplot:xlabel='lat', ylabel='Density'>
```



In [26]:

```
ds=data.fillna(20)
```

In [27]:

```
ssd=ds.head(20)
```

In [29]:

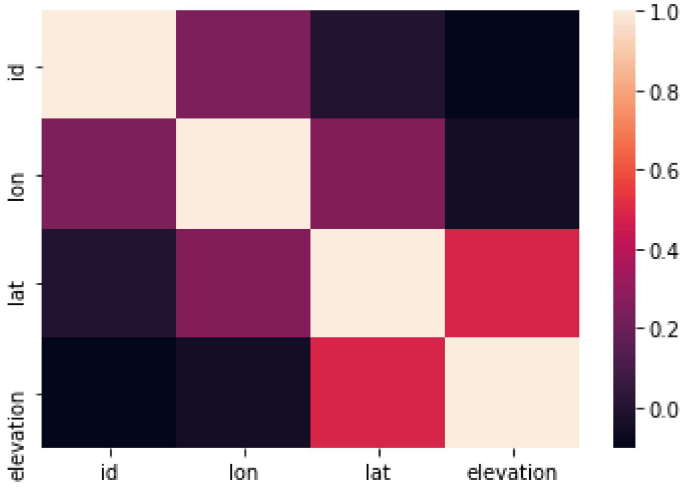
```
sd1=ssd[['id','lon', 'lat', 'elevation']]
```

In [30]:

```
sns.heatmap(ssd.corr())
```

Out[30]:

```
<AxesSubplot:>
```



In [31]:
```
x= ssd[['id','lon', 'lat']]
y=ssd['elevation']
```

In [32]:
```
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 [33]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[33]:
```
LinearRegression()
```

In [34]:
```
print(lr.intercept_)
```

```
-2532732.1379153044
```

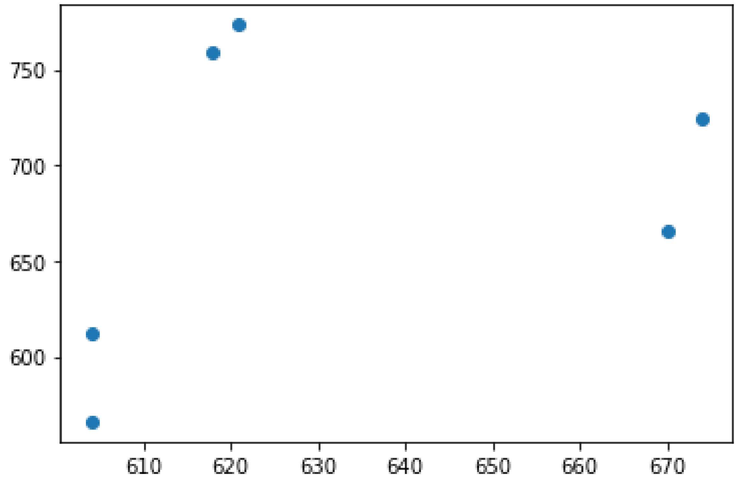
In [35]:
```
coeff= pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[35]:

| | Co-efficient |
|---|---|
| **id** | 0.088567 |
| **lon** | 394.205951 |
| **lat** | 1187.074837 |

In [36]:
```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[36]:
```
<matplotlib.collections.PathCollection at 0x20f30c99100>
```



In [37]:
```
print(lr.score(x_test,y_test))
```

```
-8.245619783922583
```

```
In [38]: lr.score(x_test,y_test)

Out[38]: -8.245619783922583
```

```
In [39]: lr.score(x_train,y_train)

Out[39]: 0.7249130048939887
```

```
In [40]: from sklearn.linear_model import Ridge,Lasso
```

```
In [41]: dr=Ridge(alpha=10)
         dr.fit(x_train,y_train)

Out[41]: Ridge(alpha=10)
```

```
In [42]: dr.score(x_test,y_test)

Out[42]: -1.3383935765207307
```

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

Out[43]: 0.0011976495708857504
```

```
In [44]: la=Lasso(alpha=10)
         la.fit(x_train,y_train)

Out[44]: Lasso(alpha=10)
```

```
In [45]: la.score(x_test,y_test)

Out[45]: -1.3466970058632453
```

```
In [46]: la.score(x_train,y_train)

Out[46]: 0.0
```

# ElasticNet

```
In [47]: from sklearn.linear_model import ElasticNet
         en=ElasticNet()
         en.fit(x_train,y_train)

Out[47]: ElasticNet()
```

```
In [48]: print(en.coef_)

         [-0.01602774  0.          0.3196718 ]
```

In [49]:

```
print(en.intercept_)
```

```
450696.08499032026
```

In [50]:

```
prediction=en.predict(x_test)
```

In [51]:

```
print(en.score(x_test,y_test))
```

```
-1.3392003029709194
```

In [52]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [53]:

```
from sklearn.linear_model import LogisticRegression
```

In [54]:

```
feature_matrix = ssd[['id','lon', 'lat']]
target_vector=ssd['elevation']
```

In [55]:

```
feature_matrix.shape
```

Out[55]:

```
(20, 3)
```

In [56]:

```
target_vector.shape
```

Out[56]:

```
(20,)
```

In [57]:

```
from sklearn.preprocessing import StandardScaler
```

In [58]:

```
fs=StandardScaler().fit_transform(feature_matrix)
```

In [59]:

```
logr= LogisticRegression()
logr.fit(fs,target_vector)
```

Out[59]:

```
LogisticRegression()
```

In [62]:

```
observation =[[1.2,2.3,3.3]]
```

In [63]:

```
prediction=logr.predict(observation)
print(prediction)
```

```
[618]
```

In [64]:

```
logr.classes_
```

Out[64]:

```
array([599, 604, 618, 621, 627, 630, 635, 642, 659, 662, 670, 674, 676,
       677, 685, 693, 698, 708, 728], dtype=int64)
```

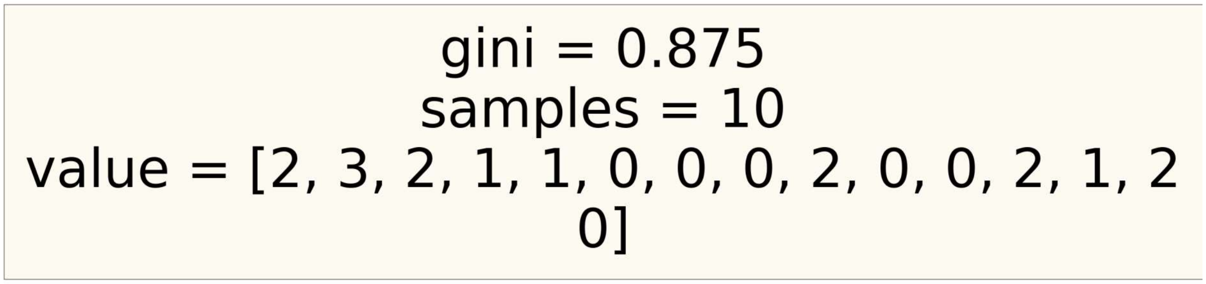
```
In [65]: logr.predict_proba(observation)[0][0]

Out[65]: 0.0033119974460056184

In [66]: ged=data[['id','lon', 'lat', 'elevation']]

In [67]: d=ged.fillna(20)

In [68]: dg=d.head(100)

In [69]: x=dg[['id','lon', 'lat']]
         y=dg['elevation']

In [70]: 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 [71]: from sklearn.ensemble import RandomForestClassifier
         rfc=RandomForestClassifier()
         rfc.fit(x_train,y_train)

Out[71]: RandomForestClassifier()

In [72]: 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 [73]: 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[73]: 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 [74]: grid_search.best_score_

Out[74]: 0.125

In [75]: rfc_best=grid_search.best_estimator_
```

In [76]:

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

Out[76]:

```
[Text(1395.0, 1087.2, 'gini = 0.875\nsamples = 10\nvalue = [2, 3, 2, 1, 1, 0,
0, 0, 2, 0, 0, 2, 1, 2\n0]')]
```



# Conclusion : LinearRegression() -2532732.1379153044 HIGH RANGE

In [ ]:

```
```