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

from google.colab import files
upload=files.upload()

<IPython.core.display.HTML object>

Saving airports.csv to airports.csv

data = pd.read_csv('/content/airports.csv')
data.drop(["id"], axis=1, inplace=True)
data.head()

  ident           type                                  name
latitude_deg  \
0   00A       heliport                  Total Rf Heliport
40.070801
1  00AA  small_airport               Aero B Ranch Airport
38.704022
2  00AK  small_airport                       Lowell Field
59.947733
3  00AL  small_airport                       Epps Airpark
34.864799
4  00AR         closed  Newport Hospital & Clinic Heliport
35.608700

   longitude_deg  elevation_ft continent iso_country iso_region
municipality  \
0     -74.933601          11.0       NaN          US      US-PA
Bensalem
1    -101.473911        3435.0       NaN          US      US-KS
Leoti
2    -151.692524         450.0       NaN          US      US-AK
Anchor Point
3     -86.770302         820.0       NaN          US      US-AL
Harvest
4     -91.254898         237.0       NaN          US      US-AR
Newport

  scheduled_service gps_code iata_code local_code home_link
wikipedia_link  \
0                no      00A       NaN        00A       NaN
NaN
1                no     00AA       NaN       00AA       NaN
NaN
2                no     00AK       NaN       00AK       NaN
NaN
3                no     00AL       NaN       00AL       NaN
```

```
NaN
4                no      NaN       NaN        NaN       NaN
NaN

  keywords
0      NaN
1      NaN
2      NaN
3      NaN
4     00AR
```

```
data.describe()
```

```
       latitude_deg  longitude_deg  elevation_ft
count  67312.000000   67312.000000  54335.000000
mean      25.945866     -31.136863   1268.620006
std       26.380436      84.227690   1624.730666
min      -90.000000    -179.876999  -1266.000000
25%       11.195161     -93.801077    208.000000
50%       35.437555     -70.799722    725.000000
75%       43.035376      18.963488   1558.000000
max       82.750000     179.975700  22000.000000
```

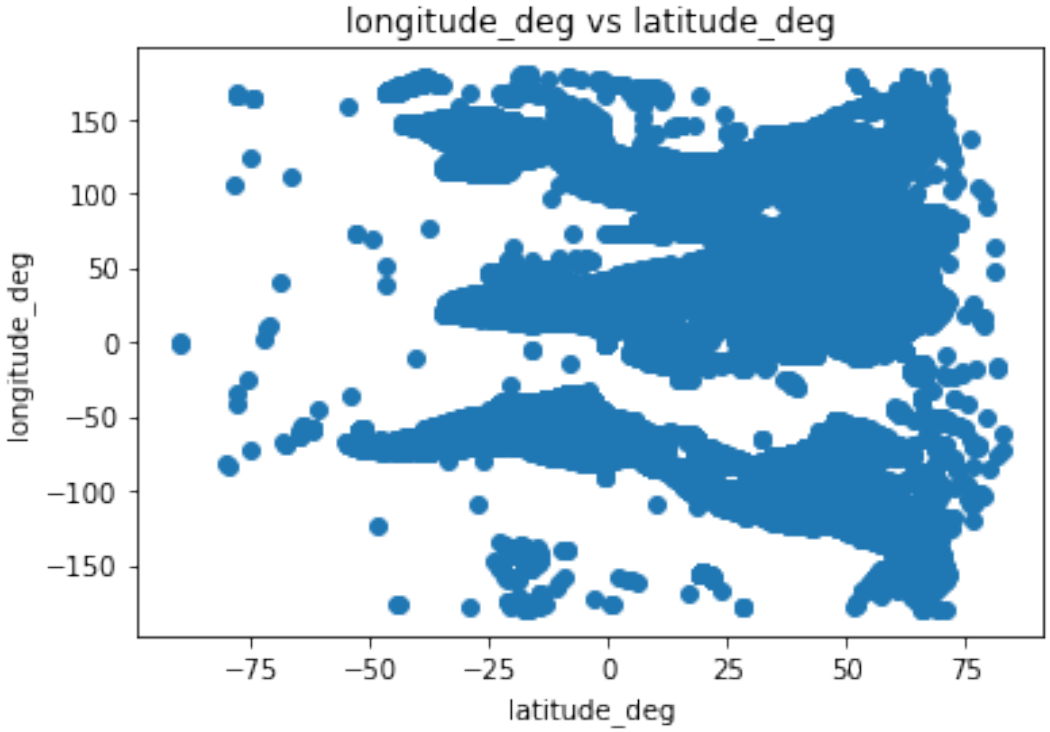
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 67312 entries, 0 to 67311
Data columns (total 17 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   ident              67311 non-null  object
 1   type               67312 non-null  object
 2   name               67312 non-null  object
 3   latitude_deg       67312 non-null  float64
 4   longitude_deg      67312 non-null  float64
 5   elevation_ft       54335 non-null  float64
 6   continent          34320 non-null  object
 7   iso_country        67055 non-null  object
 8   iso_region         67312 non-null  object
 9   municipality       61781 non-null  object
 10  scheduled_service  67312 non-null  object
 11  gps_code           42618 non-null  object
 12  iata_code          9244 non-null   object
 13  local_code         32055 non-null  object
 14  home_link          3300 non-null   object
 15  wikipedia_link     10370 non-null  object
 16  keywords           12367 non-null  object
dtypes: float64(3), object(14)
memory usage: 8.7+ MB
```

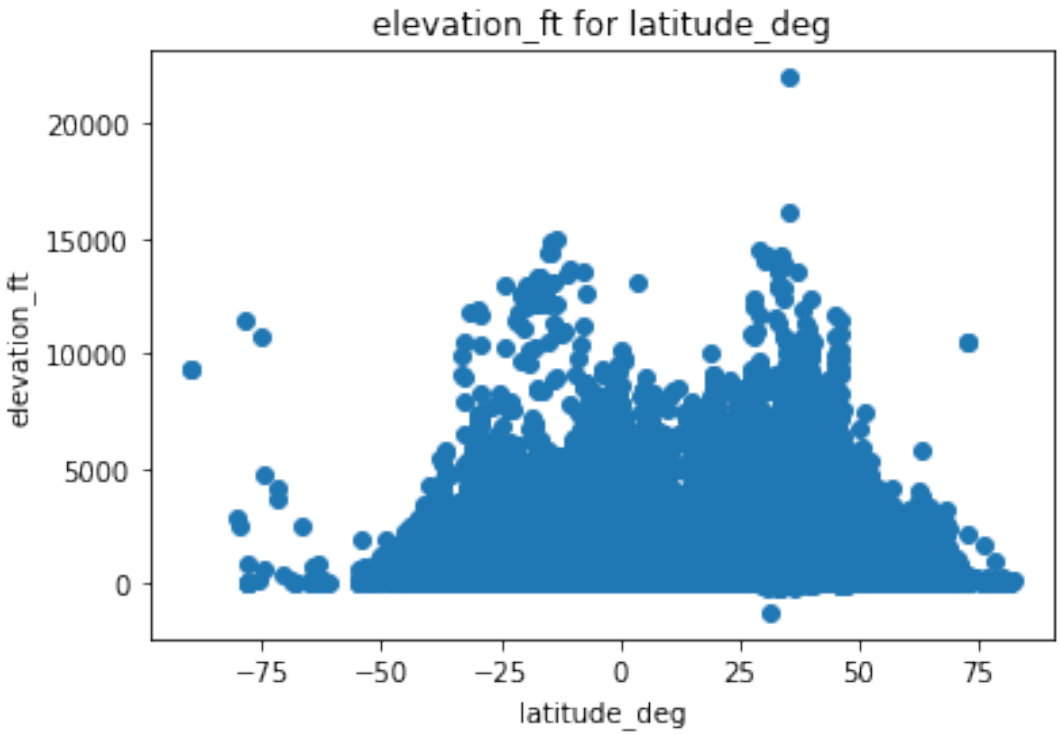
```
data.isnull().sum()
```

```
ident                     1
type                      0
name                      0
latitude_deg              0
longitude_deg             0
elevation_ft          12977
continent             32992
iso_country             257
iso_region                0
municipality           5531
scheduled_service         0
gps_code              24694
iata_code             58068
local_code            35257
home_link             64012
wikipedia_link        56942
keywords              54945
dtype: int64
```

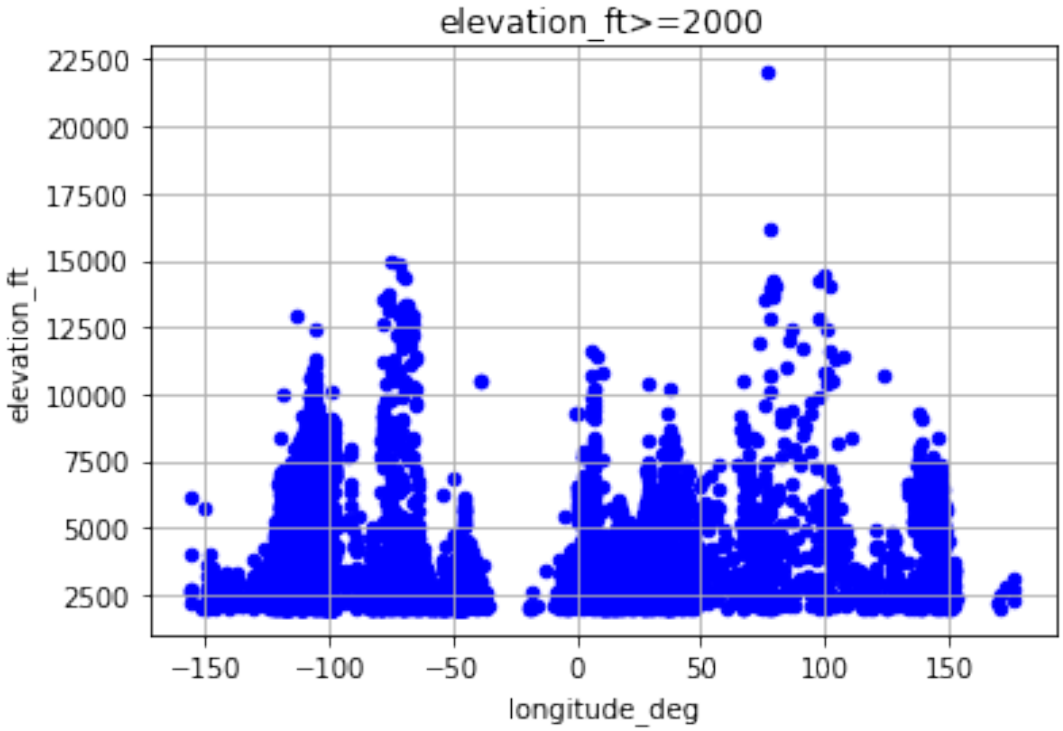
```
plt.scatter(data['latitude_deg'],data['longitude_deg'])
plt.title('longitude_deg vs latitude_deg')
plt.xlabel('latitude_deg')
plt.ylabel('longitude_deg')
plt.show()
```



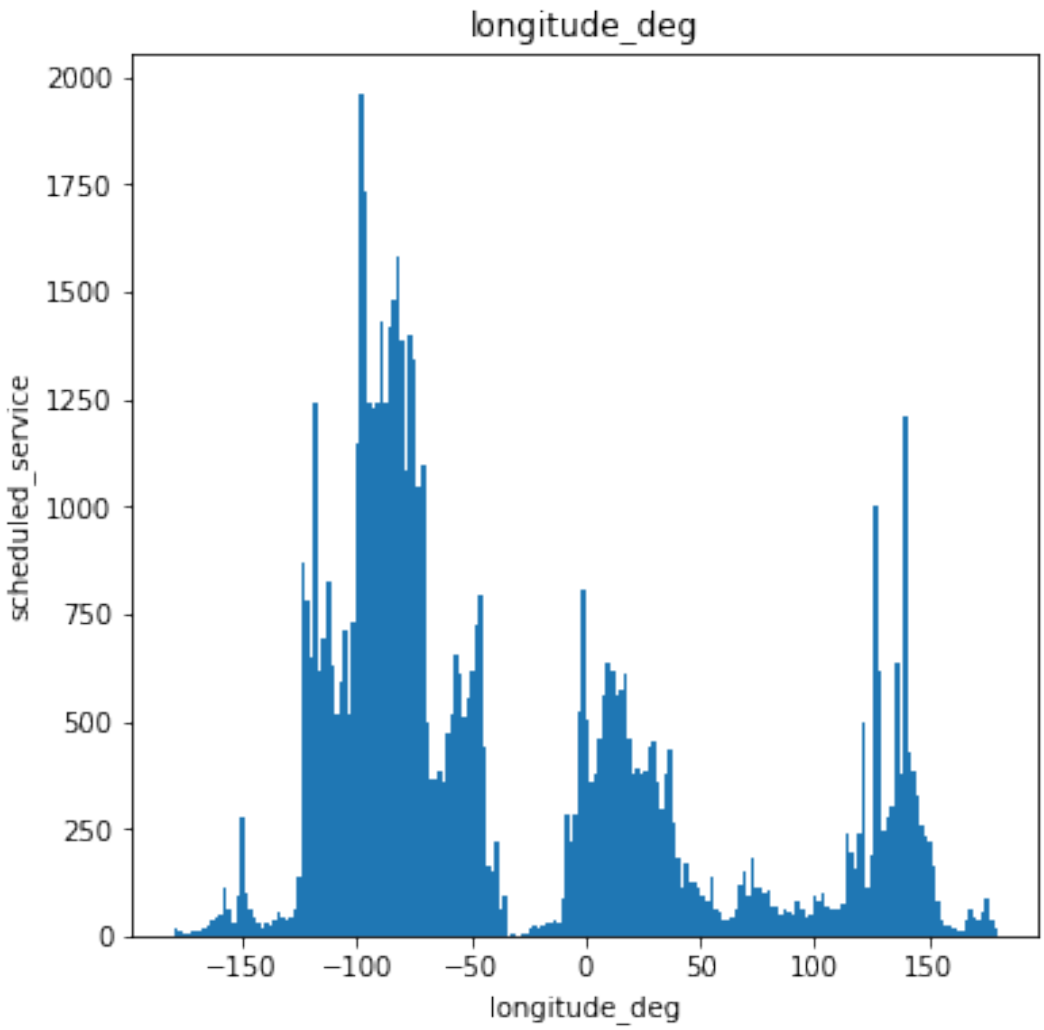
```
plt.scatter(data['latitude_deg'],data['elevation_ft'])
plt.title('elevation_ft for latitude_deg')
plt.xlabel('latitude_deg')
plt.ylabel('elevation_ft')
plt.show()
```



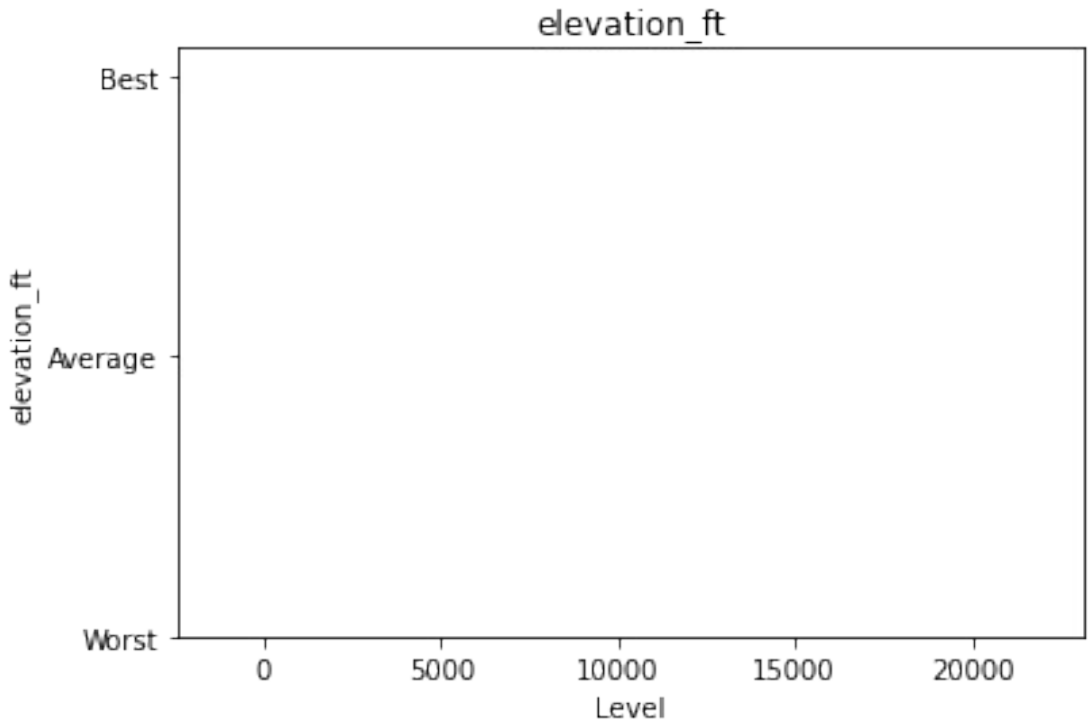

```
data[data.elevation_ft >= 2000].plot(kind='scatter',
x='longitude_deg', y='elevation_ft',color="BLUE")
plt.xlabel("longitude_deg")
plt.ylabel("elevation_ft")
plt.title("elevation_ft>=2000")
plt.grid(True)
plt.show()
```



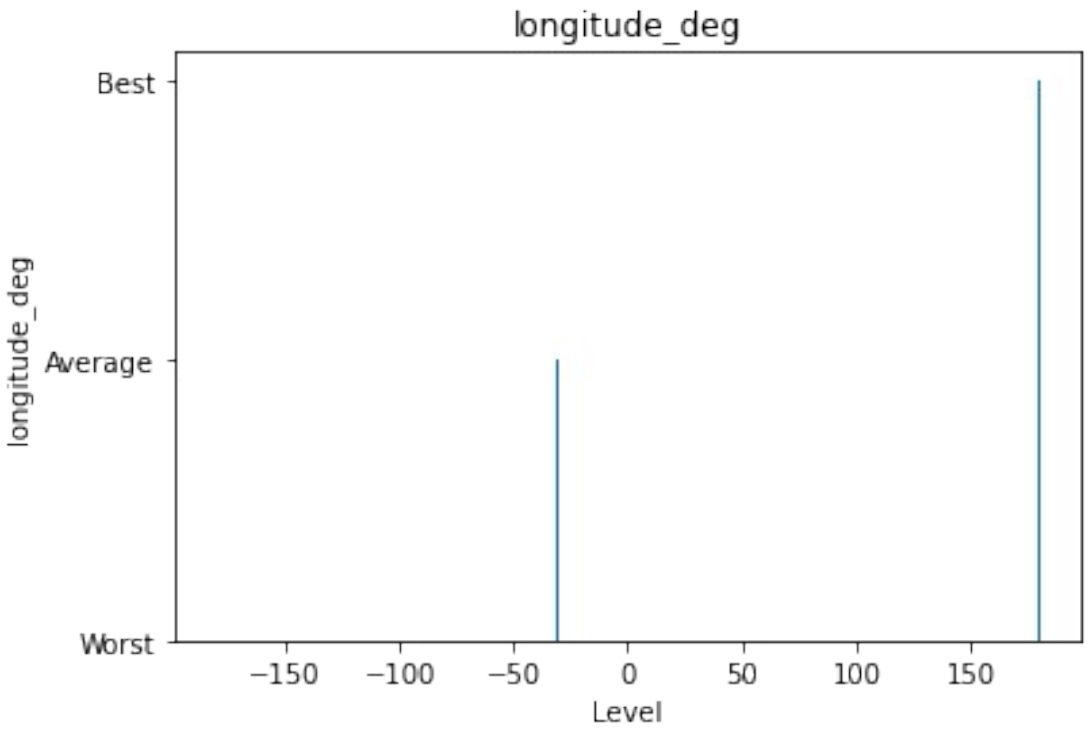

```
data["longitude_deg"].plot(kind = 'hist',bins = 200,figsize = (6,6))
plt.title("longitude_deg")
plt.xlabel("longitude_deg")
plt.ylabel("scheduled_service")
plt.show()
```



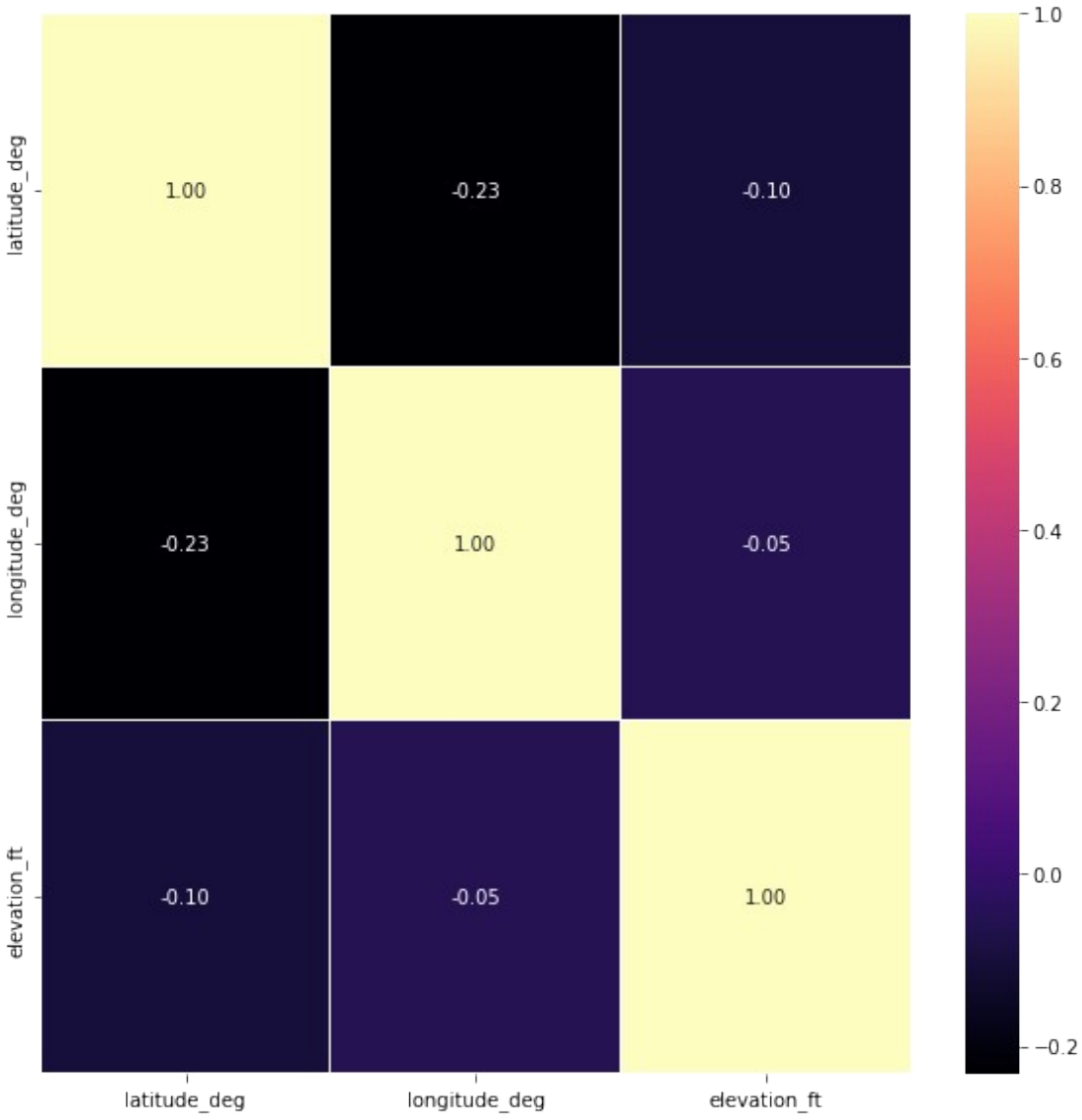

```
p =
np.array([data["elevation_ft"].min(),data["elevation_ft"].mean(),data[
"elevation_ft"].max()])
r = ["Worst","Average","Best"]
plt.bar(p,r)
plt.title("elevation_ft")
plt.xlabel("Level")
plt.ylabel("elevation_ft")
plt.show()
```




```
g =
np.array([data["longitude_deg"].min(),data["longitude_deg"].mean(),dat
a["longitude_deg"].max()])
h = ["Worst","Average","Best"]
plt.bar(g,h)
plt.title("longitude_deg")
plt.xlabel("Level")
plt.ylabel("longitude_deg")
plt.show()
```

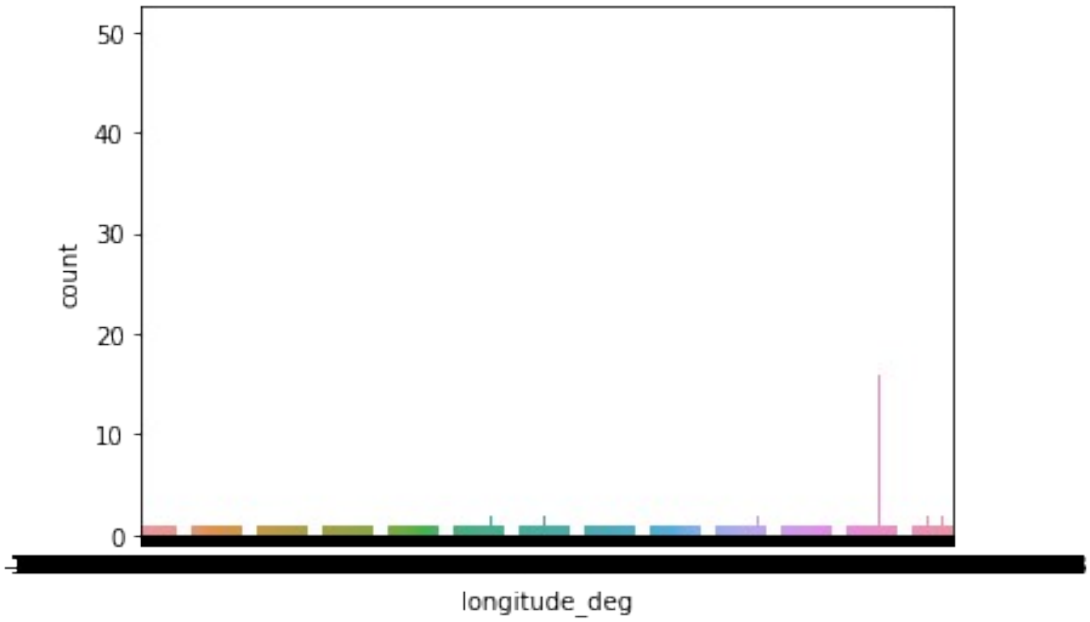



```
plt.figure(figsize=(10, 10))
sns.heatmap(data.corr(), annot=True, linewidths=0.05, fmt=
'.2f',cmap="magma")
plt.show()
```




```
data.longitude_deg.value_counts()
sns.countplot(x="longitude_deg",data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7febc56946d0>
```


50
40
30
20
10
0
count
longitude_deg