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

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  35703.000000   35703.000000  28537.000000
mean      32.146054     -33.209686   1217.982093
std       22.808783      82.179302   1529.671183
min      -89.989444    -169.511018   -210.000000
25%       30.753430     -92.494900    223.000000
50%       38.223701     -74.932899    728.000000
75%       44.843299      13.367192   1450.000000
max       82.750000     179.259167  22000.000000
```

Handling missing values

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35704 entries, 0 to 35703
Data columns (total 17 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   ident              35703 non-null  object 
 1   type               35703 non-null  object 
 2   name               35703 non-null  object 
 3   latitude_deg       35703 non-null  float64
 4   longitude_deg      35703 non-null  float64
 5   elevation_ft       28537 non-null  float64
 6   continent          16238 non-null  object 
 7   iso_country        35623 non-null  object 
 8   iso_region         35703 non-null  object 
 9   municipality       33286 non-null  object 
 10  scheduled_service  35703 non-null  object 
 11  gps_code           22685 non-null  object 
 12  iata_code          4111 non-null   object 
 13  local_code         20063 non-null  object 
 14  home_link          2172 non-null   object 
 15  wikipedia_link     5603 non-null   object 
 16  keywords           7035 non-null   object 
dtypes: float64(3), object(14)
memory usage: 4.6+ MB
```

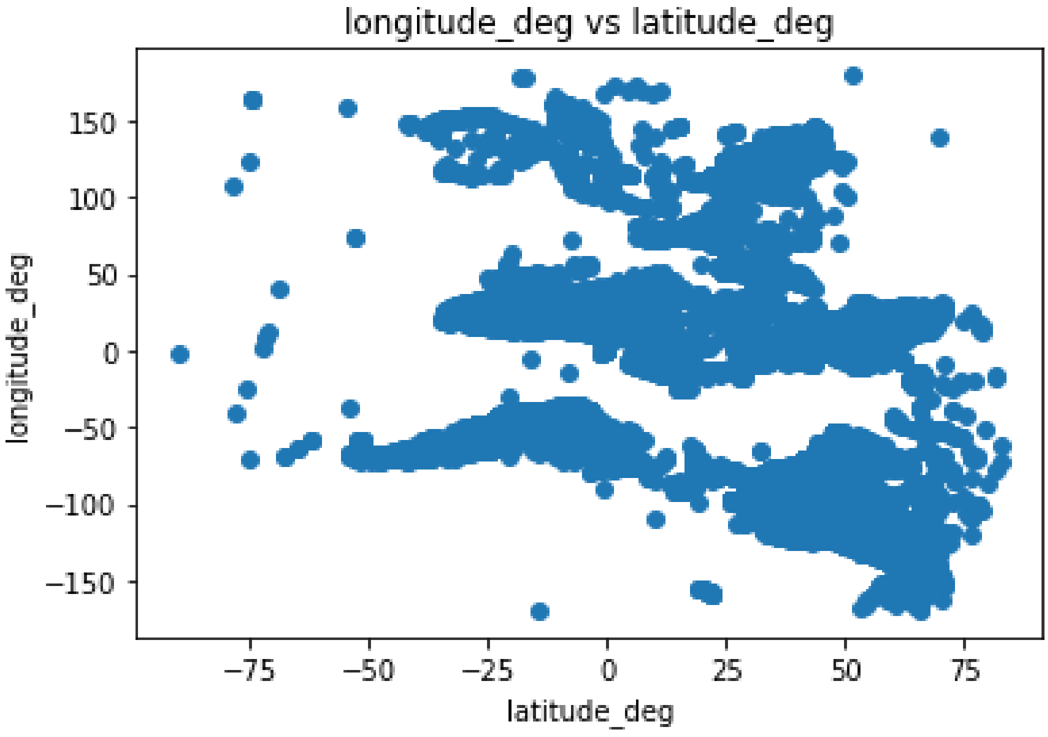
```
data.isnull().sum()
```

```
ident                    1
type                     1
name                     1
latitude_deg             1
```

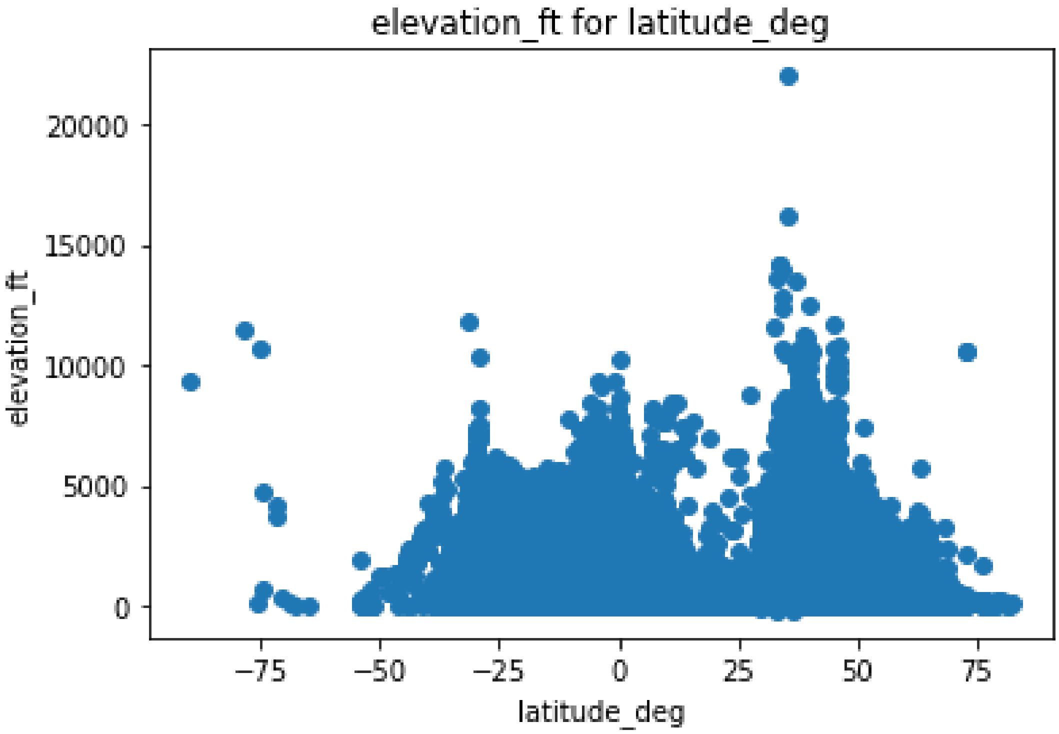
```
longitude_deg            1
elevation_ft          7167
continent            19466
iso_country             81
iso_region               1
municipality          2418
scheduled_service        1
gps_code             13019
iata_code            31593
local_code           15641
home_link            33532
wikipedia_link       30101
keywords             28669
dtype: int64
```

Data Visualization

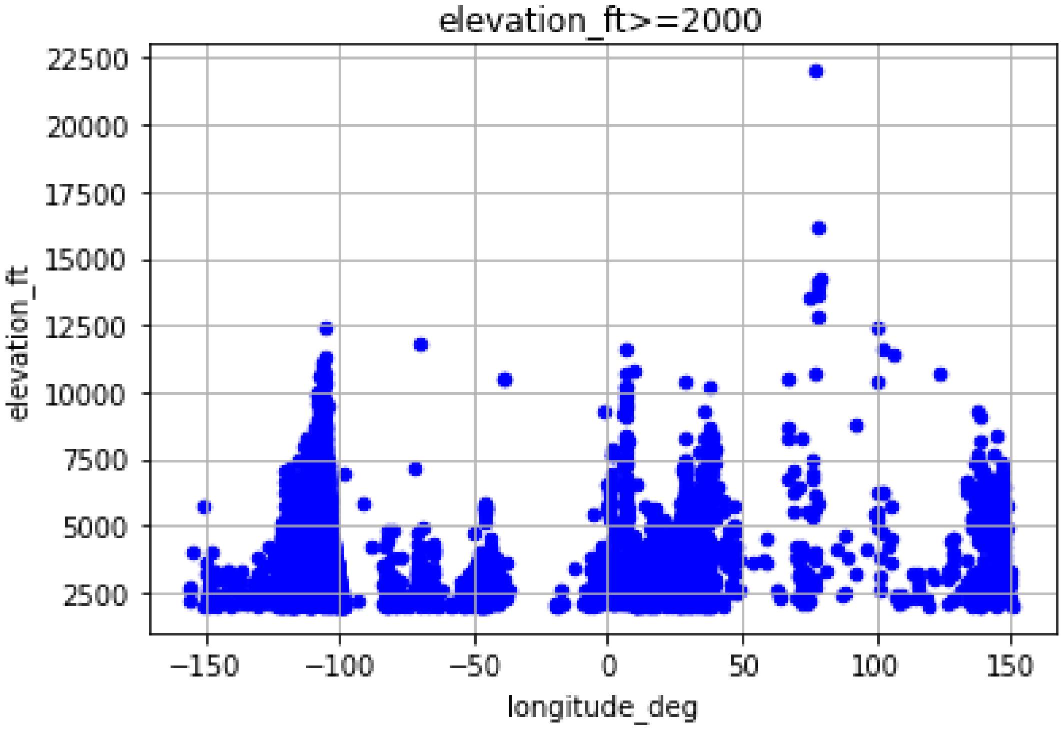
```
plt.scatter(data['latitude_deg'],data['longitude_deg'])
plt.title('longitude_deg vs latitude_deg')
plt.xlabel('latitude_deg')
plt.ylabel('longitude_deg')
plt.show()
```



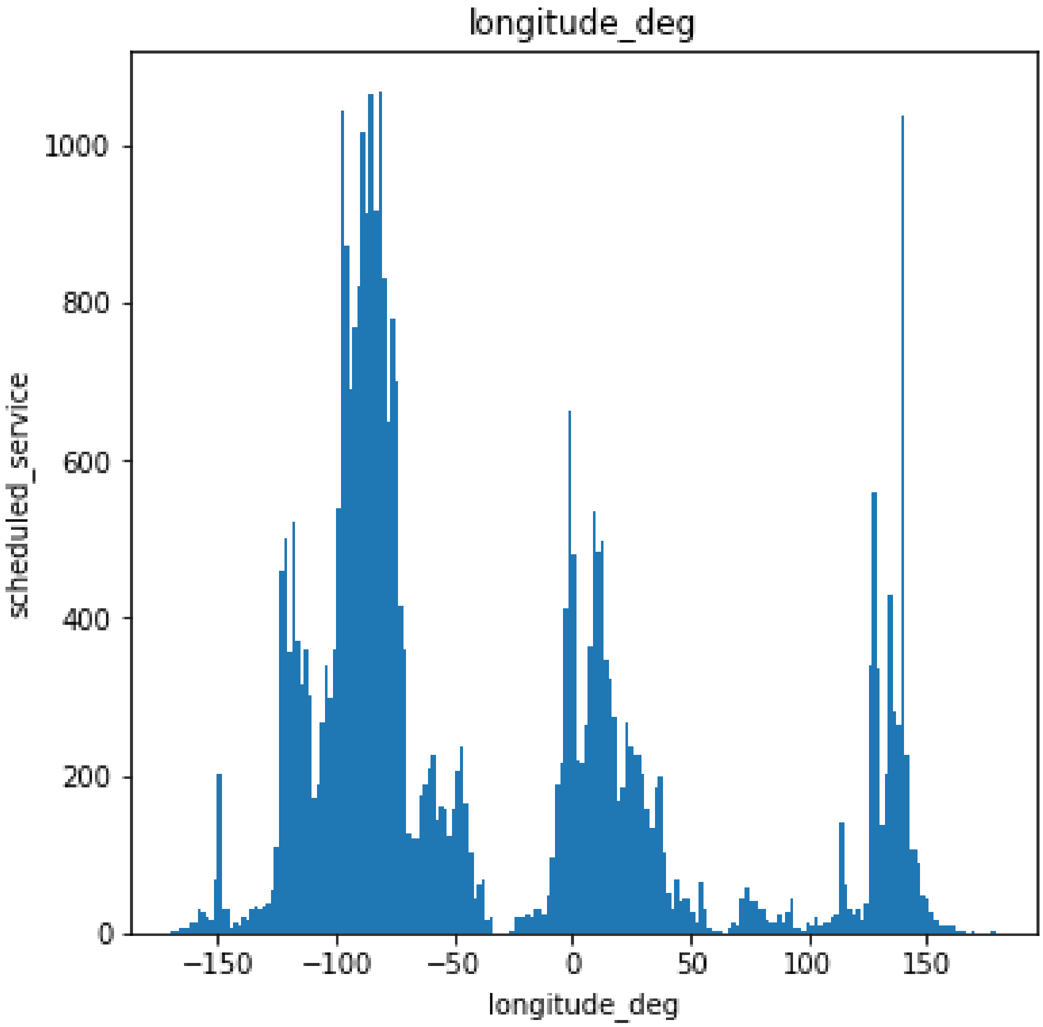
```
plt.scatter(data['latitude_deg'],data['elevation_ft'])
plt.title('elevation_ft for latitude_deg')
plt.xlabel('latitude_deg')
plt.ylabel('elevation_ft')
plt.show()
```



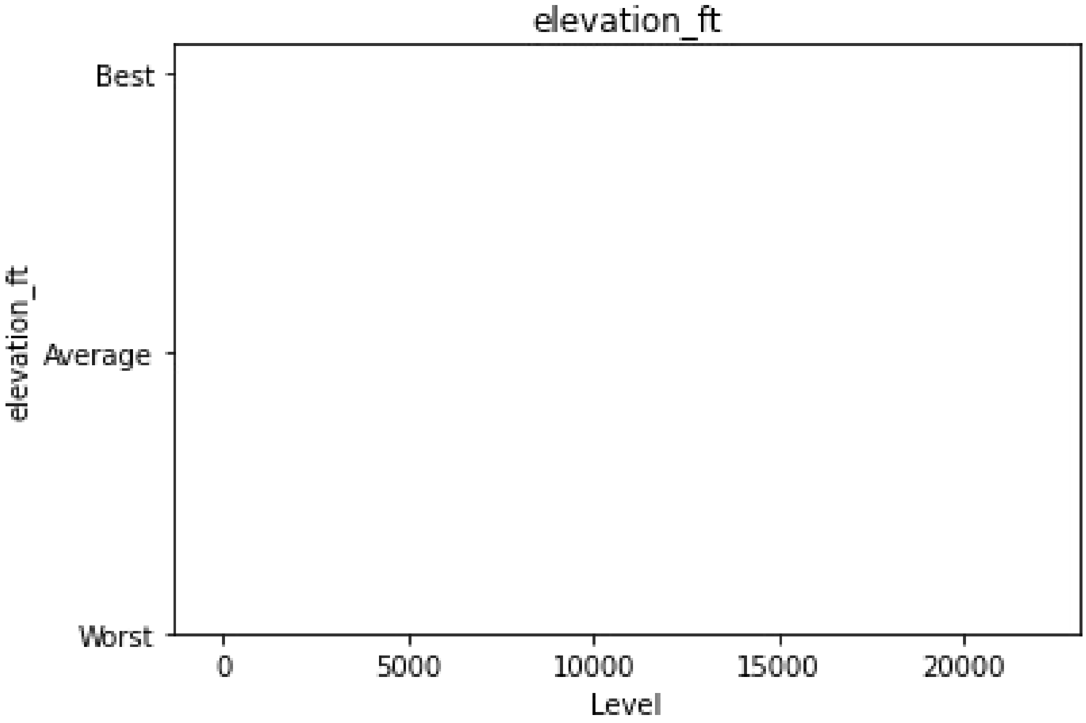
```
data[data.elevation_ft >= 2000].plot(kind='scatter',
x='longitude_deg', y='elevation_ft',color="BLUE")
plt.xlabel("longitude_deg")
plt.ylabel("elevation_ft")
plt.title("elevation_ft>=2000")
plt.grid(True)
plt.show()
```



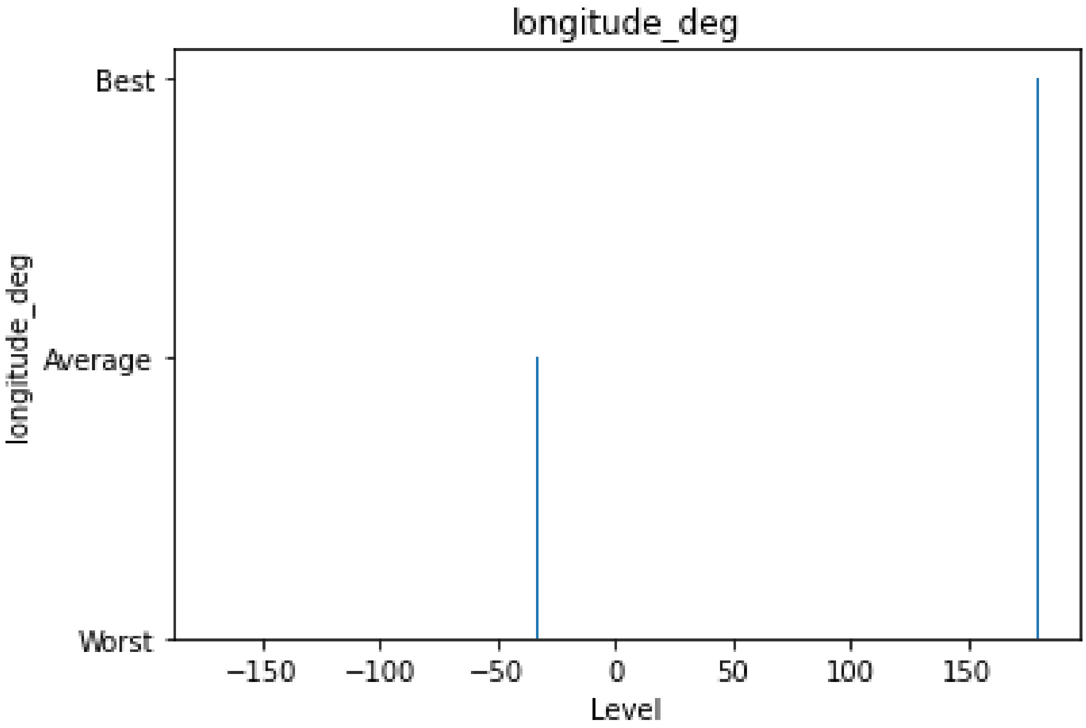
```
data["longitude_deg"].plot(kind = 'hist',bins = 200,figsize = (6,6))
plt.title("longitude_deg")
plt.xlabel("longitude_deg")
plt.ylabel("scheduled_service")
plt.show()
```



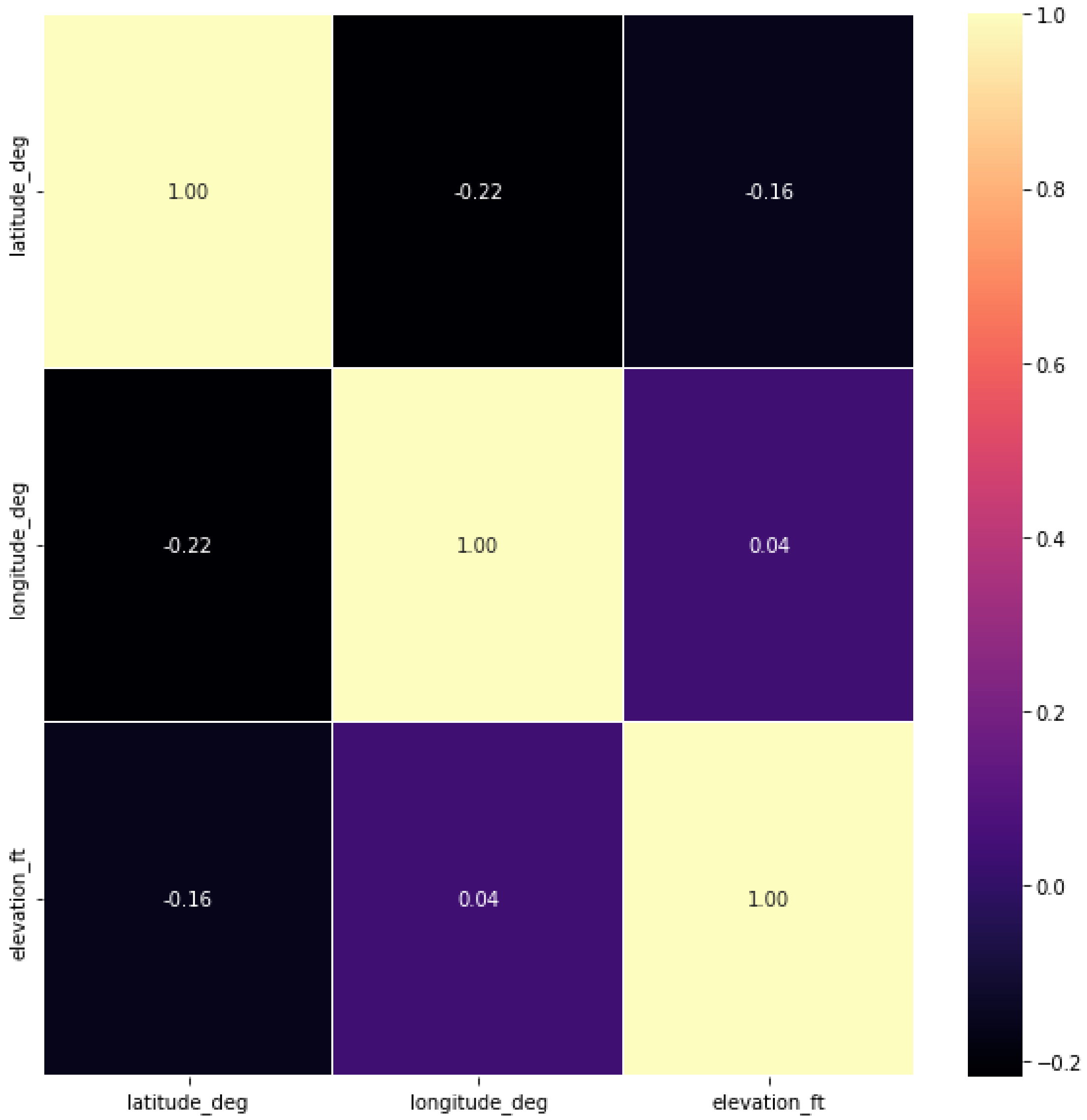
```
p =
np.array([data["elevation_ft"].min(),data["elevation_ft"].mean(),data[
"elevation_ft"].max()])
r = ["Worst","Average","Best"]
plt.bar(p,r)
plt.title("elevation_ft")
plt.xlabel("Level")
plt.ylabel("elevation_ft")
plt.show()
```



```
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()
```

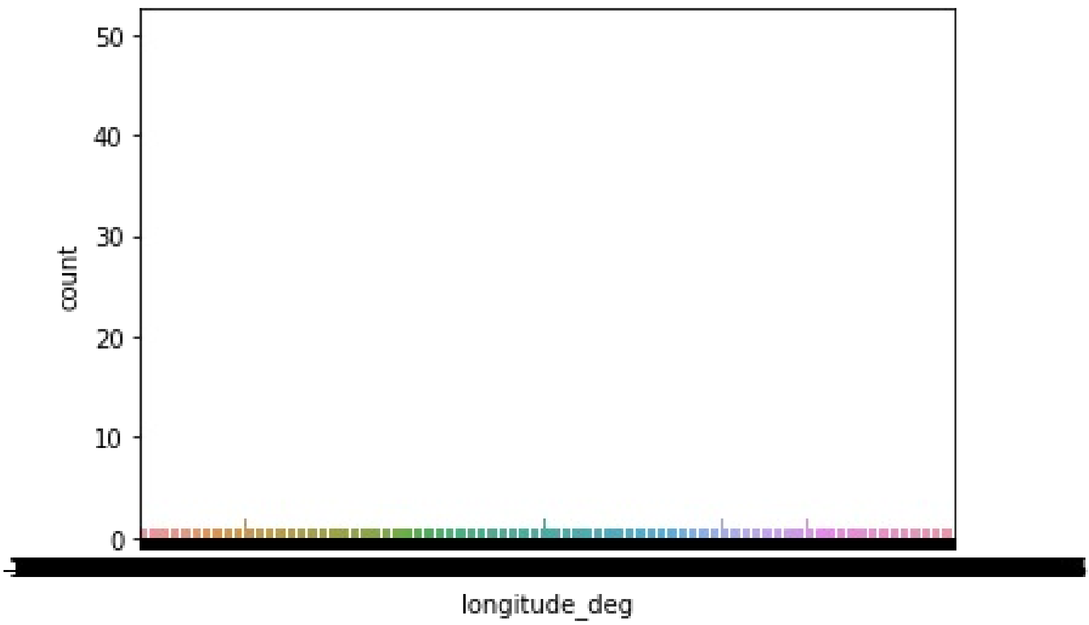


```
plt.figure(figsize=(10, 10))
sns.heatmap(data.corr(), annot=True, linewidths=0.05, fmt=
'.2f',cmap="magma")
plt.show()
```



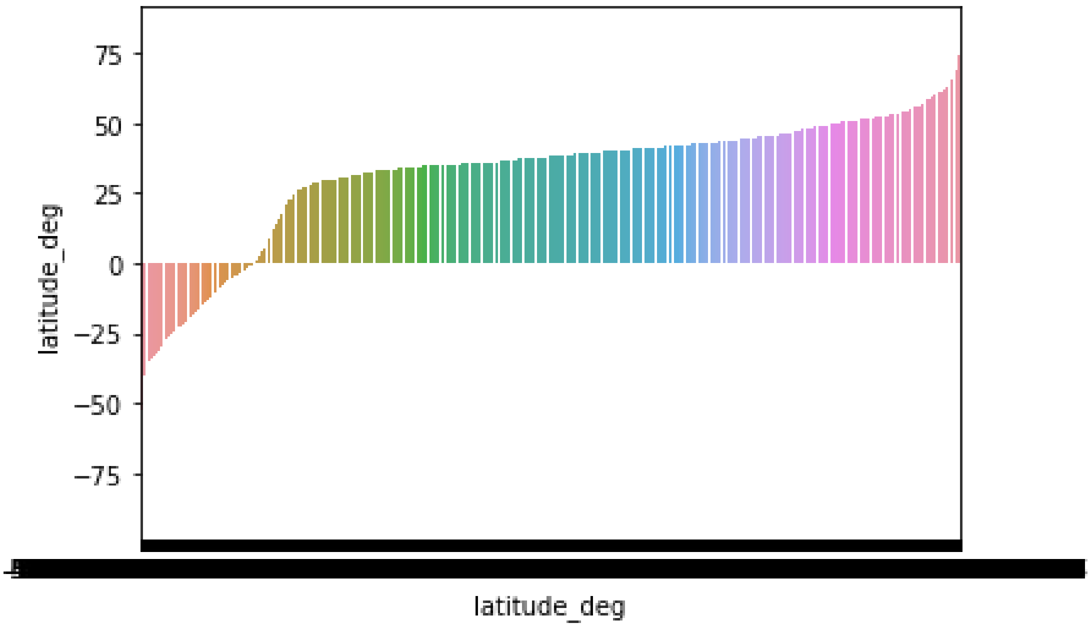
```
data.longitude_deg.value_counts()
sns.countplot(x="longitude_deg",data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe6557885d0>
```



```
sns.barplot(x="latitude_deg", y="latitude_deg", data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe619be97d0>
```


latitude_deg
latitude_deg