```
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()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

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	longitude_deg	elevation_ft	continent
0	00A	heliport	Total Rf Heliport	40.070801	-74.933601	11.0	NaN
1	00AA	small_airport	Aero B Ranch Airport	38.704022	-101.473911	3435.0	NaN
2	00AK	small_airport	Lowell Field	59.947733	-151.692524	450.0	NaN
3	00AL	small_airport	Epps Airpark	34.864799	-86.770302	820.0	NaN
4	00AR	closed	Newport Hospital & Clinic Heliport	35.608700	-91.254898	237.0	NaN

```
data.describe()
```

Handling missing values

count 67312.000000 67312.000000 54335.000000

```
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
```

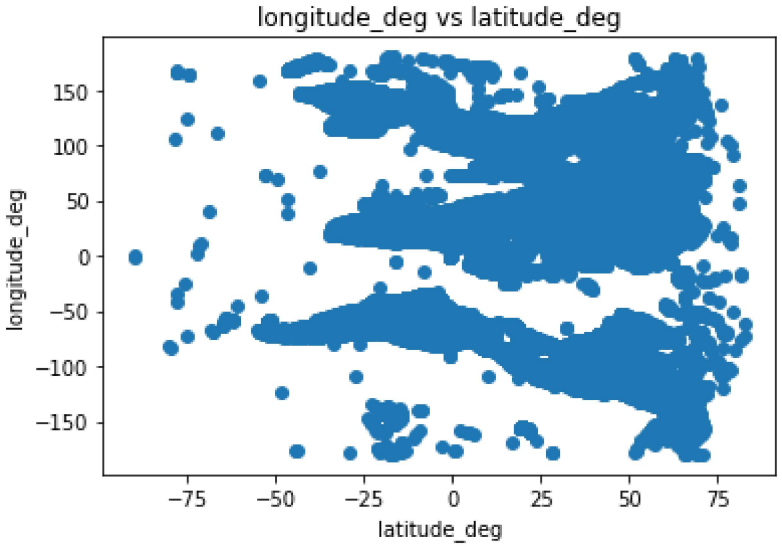
```
data.isnull().sum()
```

```
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
```

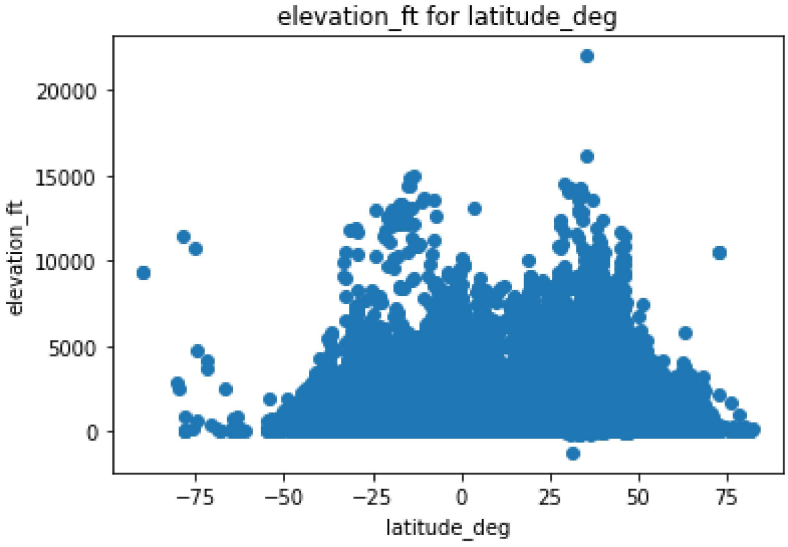
Data Visualization

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

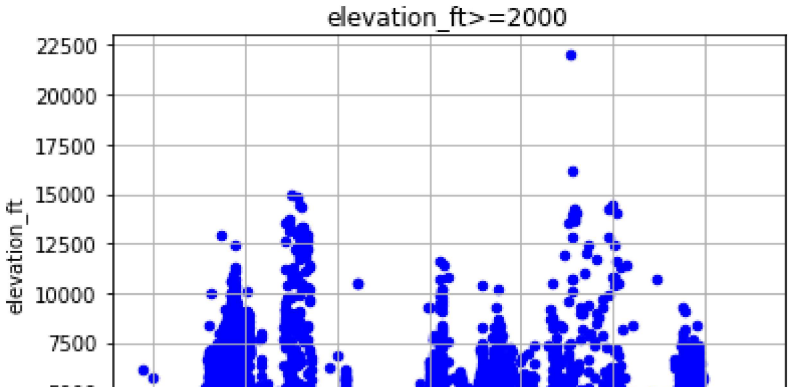
```
plt.xlabel('latitude_deg')
plt.ylabel('longitude_deg')
plt.show()
```



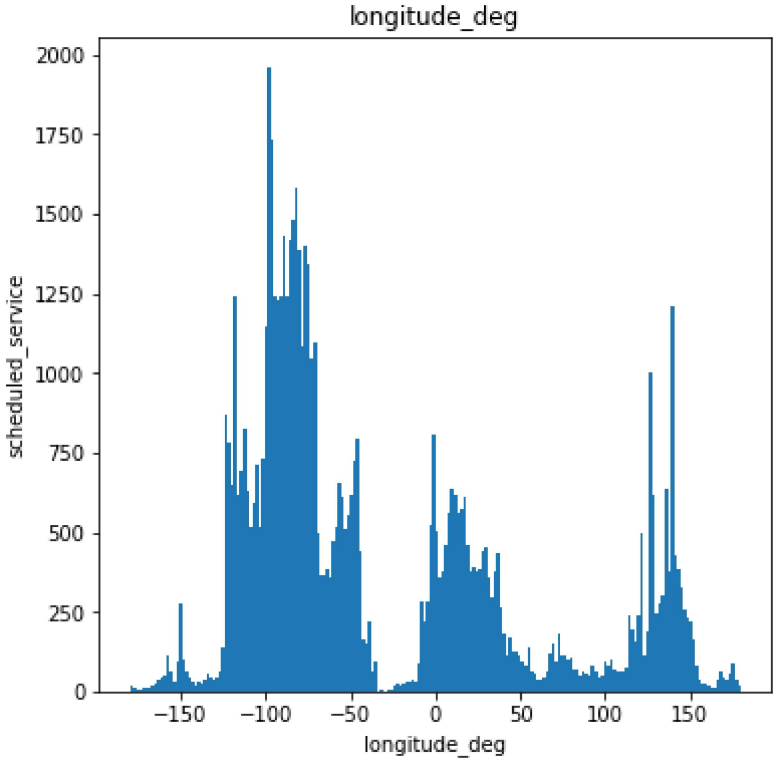

```
plt.scatter(data['latitude_deg'],data['elevation_ft'])
plt.title('elevation_ft for latitude_deg')
plt.xlabel('latitude_deg')
plt.ylabel('elevation_ft')
plt.show()
```




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

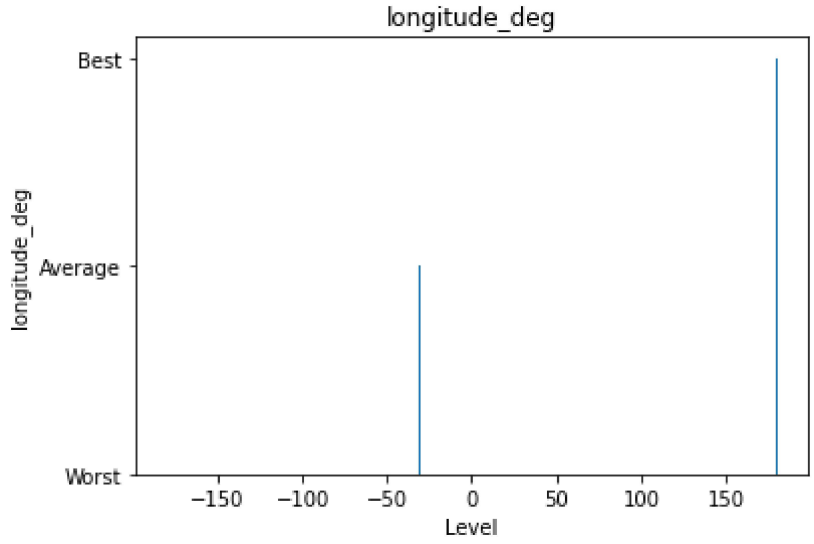



```
data["longitude_deg"].plot(kind = 'hist',bins = 200,figsize = (6,6))
plt.title("longitude_deg")
plt.xlabel("longitude_deg")
plt.ylabel("scheduled_service")
plt.show()
```

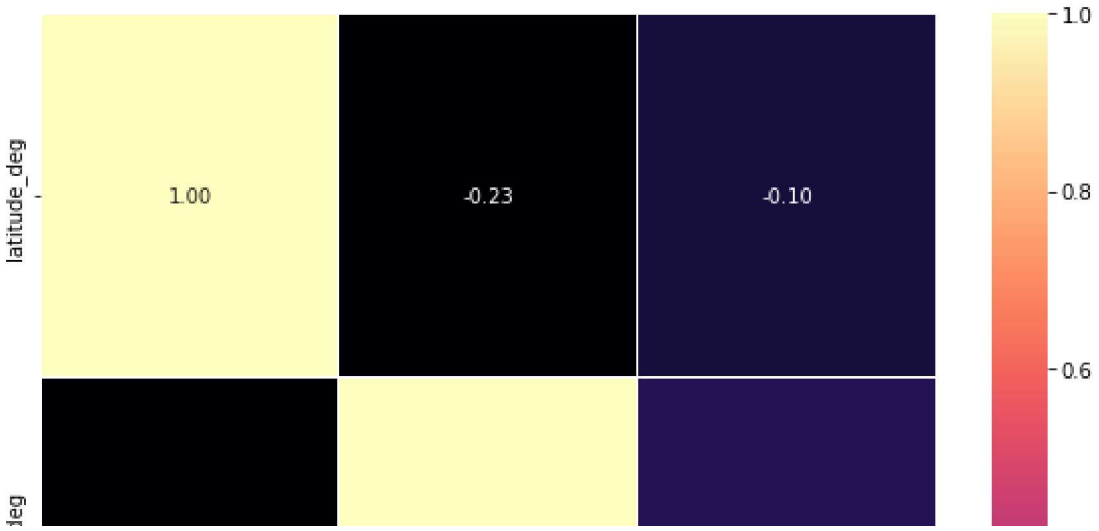



```
p = np.array([data["elevation_ft"].min(),data["elevation_ft"].mean(),data["elevation_ft"].
r = ["Worst","Average","Best"]
plt.bar(p,r)
plt.title("elevation_ft")
plt.xlabel("Level")
plt.ylabel("elevation_ft")
plt.show()
```


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

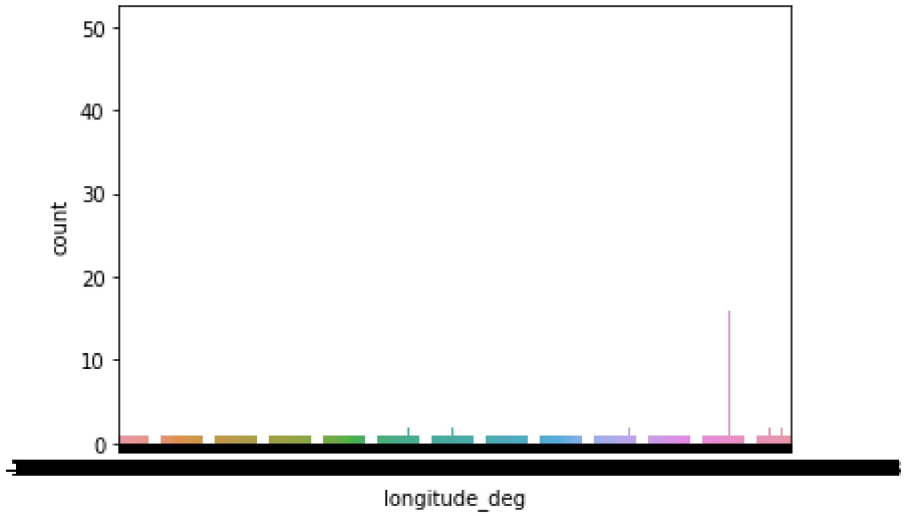



```
plt.figure(figsize=(10, 10))
sns.heatmap(data.corr(), annot=True, linewidths=0.05, fmt= '.2f',cmap="magma")
plt.show()
```




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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe2b89fd550>
```




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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe25e323a10>
```