In [1]:
```
# Task:

# Examine 2018 lightning strike data collected by the National Oceanic and Atmospheric Administration (NOAA).
# Then calculate the total number of strikes for each month and plot this information on a bar graph.
```

In [2]:
```
import pandas as pd
import numpy as np
import datetime as dt
import matplotlib.pyplot as plt
```

In [3]:
```
# Upload dataset
df = pd.read_csv(r"C:\Users\Maj Mortuza\Downloads\eda_using_basic_data_functions_in_python_dataset1.csv")
```

In [4]:
```
df.head(10)
```

Out[4]:

	date	number_of_strikes	center_point_geom
0	2018-01-03	194	POINT(-75 27)
1	2018-01-03	41	POINT(-78.4 29)
2	2018-01-03	33	POINT(-73.9 27)
3	2018-01-03	38	POINT(-73.8 27)
4	2018-01-03	92	POINT(-79 28)
5	2018-01-03	119	POINT(-78 28)
6	2018-01-03	35	POINT(-79.3 28)
7	2018-01-03	60	POINT(-79.1 28)
8	2018-01-03	41	POINT(-78.7 28)
9	2018-01-03	119	POINT(-78.6 28)

In [5]:
```
df.shape
```

Out[5]:
```
(3401012, 3)
```

In [6]:
```
# Get more information about the data, including data types of each column
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3401012 entries, 0 to 3401011
Data columns (total 3 columns):
 #   Column             Dtype 
---  ------             ----- 
 0   date               object
 1   number_of_strikes  int64 
 2   center_point_geom  object
dtypes: int64(1), object(2)
memory usage: 77.8+ MB
```

In [7]:
```
# Convert date column to datetime
df['date'] = pd.to_datetime(df['date'])
```

In [8]:
```
# Calculate days with most lightning strikes
df.groupby(['date']).sum(numeric_only=True).sort_values('number_of_strikes', ascending=False).head(10)
```

Out[8]:

date	number_of_strikes
2018-08-29	1070457
2018-08-17	969774
2018-08-28	917199
2018-08-27	824589
2018-08-30	802170
2018-08-19	786225
2018-08-18	741180
2018-08-16	734475
2018-08-31	723624
2018-08-15	673455

In [9]:
```
#Create a new `month` column
df['month'] = df['date'].dt.month
df.head()
```

Out[9]:

	date	number_of_strikes	center_point_geom	month
0	2018-01-03	194	POINT(-75 27)	1
1	2018-01-03	41	POINT(-78.4 29)	1
2	2018-01-03	33	POINT(-73.9 27)	1
3	2018-01-03	38	POINT(-73.8 27)	1
4	2018-01-03	92	POINT(-79 28)	1

In [12]:
```
# Calculate total number of strikes per month
df.groupby(['month']).sum(numeric_only=True).sort_values('number_of_strikes', ascending=False).head(12)
```

Out[12]:

month	number_of_strikes
8	15525255
7	8320400
6	6445083
5	4166726
9	3018336
2	2071315
4	1524339
10	1093962
1	860045
3	854168
11	409263
12	312097

In [13]:
```
# Convert the month number to name
df['month_name'] = df['date'].dt.month_name().str.slice(stop=3)
df.head()
```

Out[13]:

	date	number_of_strikes	center_point_geom	month	month_name
0	2018-01-03	194	POINT(-75 27)	1	Jan
1	2018-01-03	41	POINT(-78.4 29)	1	Jan
2	2018-01-03	33	POINT(-73.9 27)	1	Jan
3	2018-01-03	38	POINT(-73.8 27)	1	Jan
4	2018-01-03	92	POINT(-79 28)	1	Jan

In [14]:
```
# Create a new helper dataframe for plotting
df_by_month = df.groupby(['month', 'month_name']).sum(numeric_only=True).sort_values('month', ascending = True).head(12).reset_i
```

```
df_by_month
```

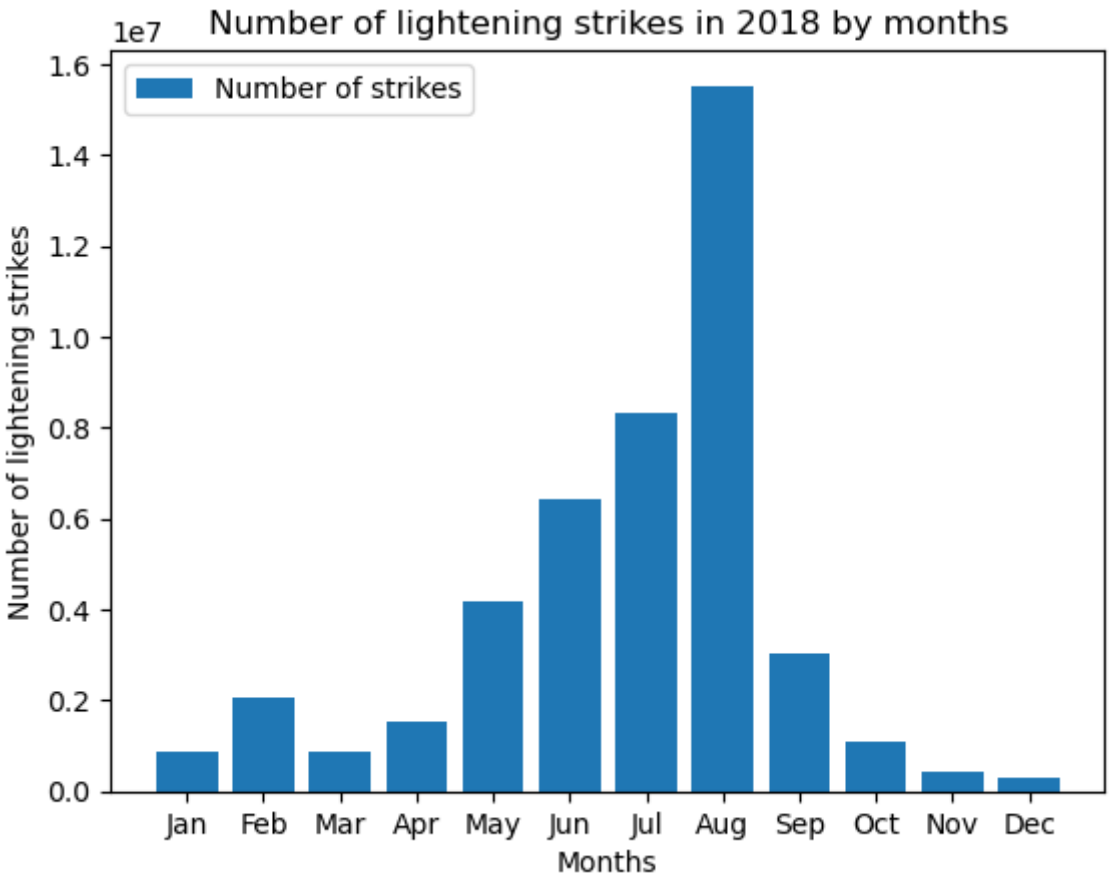
Out[14]:

	month	month_name	number_of_strikes
0	1	Jan	860045
1	2	Feb	2071315
2	3	Mar	854168
3	4	Apr	1524339
4	5	May	4166726
5	6	Jun	6445083
6	7	Jul	8320400
7	8	Aug	15525255
8	9	Sep	3018336
9	10	Oct	1093962
10	11	Nov	409263
11	12	Dec	312097

In [15]:

```
# Now create a bar chart for the viz
plt.bar(x=df_by_month['month_name'], height=df_by_month['number_of_strikes'], label="Number of strikes")

plt.xlabel("Months")
plt.ylabel("Number of lightening strikes")
plt.title("Number of lightening strikes in 2018 by months")
plt.legend()
plt.show()
```


Number of lightening strikes in 2018 by months
1e7
Number of strikes
Number of lightening strikes
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Months