In [18]:
```
import pandas as pd
```

In [71]:
```
meteorites = pd.read_csv('Meteorite_Landings.csv')
meteorites
```

Out[71]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat
0	Aachen	1	Valid	L5	21.0	Fell	01/01/1880 12:00:00 AM	50.77500
1	Aarhus	2	Valid	H6	720.0	Fell	01/01/1951 12:00:00 AM	56.18333
2	Abee	6	Valid	EH4	107000.0	Fell	01/01/1952 12:00:00 AM	54.21667
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	01/01/1976 12:00:00 AM	16.88333
4	Achiras	370	Valid	L6	780.0	Fell	01/01/1902 12:00:00 AM	-33.16667
...	...	...	...	...	...	...	...	...
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	01/01/1990 12:00:00 AM	29.03700
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	01/01/1999 12:00:00 AM	13.78333
45713	Zlin	30410	Valid	H4	3.3	Found	01/01/1939 12:00:00 AM	49.25000
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	01/01/2003 12:00:00 AM	49.78917
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	01/01/1976 12:00:00 AM	33.98333

45716 rows × 10 columns

In [35]:
```
meteorites['name']
meteorites.name
```

Out[35]:
```
0      Aachen
1      Aarhus
2        Abee
3    Acapulco
4     Achiras
Name: name, dtype: object
```

In [26]:
```
meteorites.columns
```

Out[26]:
```
Index(['name', 'id', 'nametype', 'recclass', 'mass (g)', 'fall', 'year',
       'reclat', 'reclong', 'GeoLocation'],
      dtype='object')
```

In [27]:
```
meteorites.index
```

Out[27]:
```
RangeIndex(start=0, stop=5, step=1)
```

In [54]:
```
import requests

response = requests.get('https://data.nasa.gov/resource/gh4g-9sfh.json',
                        params = {'$limit': 50_000})

if response.ok:
    payload = response.json()
else:
    print(f'Request was not successful and returned code: {response.status_code}.')
    payload = None
```

In [58]:
```
payload[:5]
```

Out[58]:

```
[{'name': 'Aachen',
  'id': '1',
  'nametype': 'Valid',
  'recclass': 'L5',
  'mass': '21',
  'fall': 'Fell',
  'year': '1880-01-01T00:00:00.000',
  'reclat': '50.775000',
  'reclong': '6.083330',
  'geolocation': {'latitude': '50.775', 'longitude': '6.08333'}},
 {'name': 'Aarhus',
  'id': '2',
  'nametype': 'Valid',
  'recclass': 'H6',
  'mass': '720',
  'fall': 'Fell',
  'year': '1951-01-01T00:00:00.000',
  'reclat': '56.183330',
  'reclong': '10.233330',
  'geolocation': {'latitude': '56.18333', 'longitude': '10.23333'}},
 {'name': 'Abee',
  'id': '6',
  'nametype': 'Valid',
  'recclass': 'EH4',
  'mass': '107000',
  'fall': 'Fell',
  'year': '1952-01-01T00:00:00.000',
  'reclat': '54.216670',
  'reclong': '-113.000000',
  'geolocation': {'latitude': '54.21667', 'longitude': '-113.0'}},
 {'name': 'Acapulco',
  'id': '10',
  'nametype': 'Valid',
  'recclass': 'Acapulcoite',
  'mass': '1914',
  'fall': 'Fell',
  'year': '1976-01-01T00:00:00.000',
  'reclat': '16.883330',
  'reclong': '-99.900000',
  'geolocation': {'latitude': '16.88333', 'longitude': '-99.9'}},
 {'name': 'Achiras',
  'id': '370',
  'nametype': 'Valid',
  'recclass': 'L6',
  'mass': '780',
  'fall': 'Fell',
  'year': '1902-01-01T00:00:00.000',
  'reclat': '-33.166670',
  'reclong': '-64.950000',
  'geolocation': {'latitude': '-33.16667', 'longitude': '-64.95'}}]
```


In [62]:
```
df = pd.DataFrame(payload)
df.head(3)
```

Out[62]:

	name	id	nametype	recclass	mass	fall	year	reclat	reclong	g
0	Aachen	1	Valid	L5	21	Fell	1880-01-01T00:00:00.000	50.775000	6.083330	
1	Aarhus	2	Valid	H6	720	Fell	1951-01-01T00:00:00.000	56.183330	10.233330	
2	Abee	6	Valid	EH4	107000	Fell	1952-01-01T00:00:00.000	54.216670	-113.000000	

In [60]:
```
meteorites
```

Out[60]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong	G
0	Aachen	1	Valid	L5	21	Fell	01/01/1880 12:00:00 AM	50.77500	6.08333	
1	Aarhus	2	Valid	H6	720	Fell	01/01/1951 12:00:00 AM	56.18333	10.23333	
2	Abee	6	Valid	EH4	107000	Fell	01/01/1952 12:00:00 AM	54.21667	-113.00000	
3	Acapulco	10	Valid	Acapulcoite	1914	Fell	01/01/1976 12:00:00 AM	16.88333	-99.90000	
4	Achiras	370	Valid	L6	780	Fell	01/01/1902 12:00:00 AM	-33.16667	-64.95000	

In [68]:
```
meteorites.shape #attribute of meteorites
```

Out[68]:
```
(5, 10)
```

In [67]:
```
meteorites.columns
```

Out[67]:
```
Index(['name', 'id', 'nametype', 'recclass', 'mass (g)', 'fall', 'year',
       'reclat', 'reclong', 'GeoLocation'],
      dtype='object')
```

In [69]:
```
meteorites.dtypes #attribute of meteorites
```

Out[69]:
```
name            object
id               int64
nametype        object
recclass        object
mass (g)         int64
fall            object
year            object
reclat         float64
reclong        float64
GeoLocation     object
dtype: object
```

In [72]:
```
meteorites.head(10)
```

Out[72]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong
0	Aachen	1	Valid	L5	21.0	Fell	01/01/1880 12:00:00 AM	50.77500	6.08333
1	Aarhus	2	Valid	H6	720.0	Fell	01/01/1951 12:00:00 AM	56.18333	10.23333
2	Abee	6	Valid	EH4	107000.0	Fell	01/01/1952 12:00:00 AM	54.21667	-113.00000
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	01/01/1976 12:00:00 AM	16.88333	-99.90000
4	Achiras	370	Valid	L6	780.0	Fell	01/01/1902 12:00:00 AM	-33.16667	-64.95000
5	Adhi Kot	379	Valid	EH4	4239.0	Fell	01/01/1919 12:00:00 AM	32.10000	71.80000
6	Adzhi-Bogdo (stone)	390	Valid	LL3-6	910.0	Fell	01/01/1949 12:00:00 AM	44.83333	95.16667
7	Agen	392	Valid	H5	30000.0	Fell	01/01/1814 12:00:00 AM	44.21667	0.61667
8	Aguada	398	Valid	L6	1620.0	Fell	01/01/1930 12:00:00 AM	-31.60000	-65.23333
9	Aguila Blanca	417	Valid	L	1440.0	Fell	01/01/1920 12:00:00 AM	-30.86667	-64.55000

In [73]:
```
meteorites.tail()
```

Out[73]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	r
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	01/01/1990 12:00:00 AM	29.03700	17
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	01/01/1999 12:00:00 AM	13.78333	8
45713	Zlin	30410	Valid	H4	3.3	Found	01/01/1939 12:00:00 AM	49.25000	17
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	01/01/2003 12:00:00 AM	49.78917	41
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	01/01/1976 12:00:00 AM	33.98333	-115

In [74]:
```
meteorites.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45716 entries, 0 to 45715
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   name         45716 non-null  object 
 1   id           45716 non-null  int64  
 2   nametype     45716 non-null  object 
 3   recclass     45716 non-null  object 
 4   mass (g)     45585 non-null  float64
 5   fall         45716 non-null  object 
 6   year         45425 non-null  object 
 7   reclat       38401 non-null  float64
 8   reclong      38401 non-null  float64
 9   GeoLocation  38401 non-null  object 
dtypes: float64(3), int64(1), object(6)
memory usage: 3.5+ MB
```

In [79]:
```
meteorites[['name','recclass']]
```

Out[79]:

	name	recclass
0	Aachen	L5
1	Aarhus	H6
2	Abee	EH4
3	Acapulco	Acapulcoite
4	Achiras	L6
...	...	...
45711	Zillah 002	Eucrite
45712	Zinder	Pallasite, ungrouped
45713	Zlin	H4
45714	Zubkovsky	L6
45715	Zulu Queen	L3.7

45716 rows × 2 columns

In [86]:
```
meteorites.iloc[100:104, [0,3,4,6]]
```

Out[86]:

	name	recclass	mass (g)	year
100	Benton	LL6	2840.0	01/01/1949 12:00:00 AM
101	Berduc	L6	270.0	01/01/2008 12:00:00 AM
102	Béréba	Eucrite-mmict	18000.0	01/01/1924 12:00:00 AM
103	Berlanguillas	L6	1440.0	01/01/1811 12:00:00 AM

In [89]:
```
meteorites.loc[100:104,'mass (g)':'year']
```

Out[89]:

	mass (g)	fall	year
100	2840.0	Fell	01/01/1949 12:00:00 AM
101	270.0	Fell	01/01/2008 12:00:00 AM
102	18000.0	Fell	01/01/1924 12:00:00 AM
103	1440.0	Fell	01/01/1811 12:00:00 AM
104	960.0	Fell	01/01/2004 12:00:00 AM

In [94]:
```
meteorites.iloc[-1, -1]
```

Out[94]:
```
'(33.98333, -115.68333)'
```

In [99]:
```
(meteorites['mass (g)']>50) & (meteorites.fall=='Found')
```

Out[99]:
```
0        False
1        False
2        False
3        False
4        False
         ...
45711     True
45712    False
45713    False
45714     True
45715     True
Length: 45716, dtype: bool
```

In [101…
```
meteorites[(meteorites['mass (g)']>50) & (meteorites.fall=='Found')]
```

Out[101…

	name	id	nametype	recclass	mass (g)	fall	year	reclat	
37	Northwest Africa 5815	50693	Valid	L5	256.80	Found	NaN	0.00000	
757	Dominion Range 03239	32591	Valid	L6	69.50	Found	01/01/2002 12:00:00 AM	NaN	
804	Dominion Range 03240	32592	Valid	LL5	290.90	Found	01/01/2002 12:00:00 AM	NaN	
1111	Abajo	4	Valid	H5	331.00	Found	01/01/1982 12:00:00 AM	26.80000	-1
1112	Abar al' Uj 001	51399	Valid	H3.8	194.34	Found	01/01/2008 12:00:00 AM	22.72192	
…	…	…	…	…	…	…	…	…	
45709	Zhongxiang	30406	Valid	Iron	100000.00	Found	01/01/1981 12:00:00 AM	31.20000	1
45710	Zillah 001	31355	Valid	L6	1475.00	Found	01/01/1990 12:00:00 AM	29.03700	
45711	Zillah 002	31356	Valid	Eucrite	172.00	Found	01/01/1990 12:00:00 AM	29.03700	
45714	Zubkovsky	31357	Valid	L6	2167.00	Found	01/01/2003 12:00:00 AM	49.78917	
45715	Zulu Queen	30414	Valid	L3.7	200.00	Found	01/01/1976 12:00:00 AM	33.98333	-1

18854 rows × 10 columns

In [105…

```
meteorites.query("`mass (g)` > 1e6 and fall=='Fell'")
```

Out[105…

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclon
29	Allende	2278	Valid	CV3	2000000.0	Fell	01/01/1969 12:00:00 AM	26.96667	-105.3166
419	Jilin	12171	Valid	H5	4000000.0	Fell	01/01/1976 12:00:00 AM	44.05000	126.1666
506	Kunya-Urgench	12379	Valid	H5	1100000.0	Fell	01/01/1998 12:00:00 AM	42.25000	59.2000
707	Norton County	17922	Valid	Aubrite	1100000.0	Fell	01/01/1948 12:00:00 AM	39.68333	-99.8666
920	Sikhote-Alin	23593	Valid	Iron, IIAB	23000000.0	Fell	01/01/1947 12:00:00 AM	46.16000	134.6533

In [107…
```
meteorites.fall.value_counts()
```

Out[107…
```
fall
Found    44609
Fell      1107
Name: count, dtype: int64
```

In [110…
```
meteorites.value_counts(subset = ['nametype', 'fall'], normalize = False)#false ret
```

Out[110…
```
nametype  fall
Valid     Found    44534
          Fell      1107
Relict    Found       75
Name: count, dtype: int64
```

In [111…
```
meteorites['mass (g)'].quantile([0.01, 0.05, 0.5, 0.95, 0.99])
```

Out[111…
```
0.01        0.44
0.05        1.10
0.50       32.60
0.95     4000.00
0.99    50600.00
Name: mass (g), dtype: float64
```

In [119…
```
type(meteorites['mass (g)'].mean())
```

Out[119…
```
numpy.float64
```

In [121…
```
meteorites['mass (g)'].median()
```

Out[121…
```
32.6
```

In [122…
```
meteorites['mass (g)'].max()
```

Out[122…
```
60000000.0
```

In [129…
```
meteorites.loc[meteorites['mass (g)'].idxmax()]
```

Out[129…
```
name                           Hoba
id                            11890
nametype                      Valid
recclass                  Iron, IVB
mass (g)                 60000000.0
fall                          Found
year           01/01/1920 12:00:00 AM
reclat                      -19.58333
reclong                      17.91667
GeoLocation     (-19.58333, 17.91667)
Name: 16392, dtype: object
```

In [131…
```
meteorites.recclass.nunique()
```

Out[131…
```
466
```

In [132…
```
meteorites.name.nunique()
```

Out[132…
```
45716
```

In [136…
```
meteorites.recclass.unique()[:4]
```

Out[136…
```
array(['L5', 'H6', 'EH4', 'Acapulcoite'], dtype=object)
```

In [137…
```
meteorites.describe()
```

Out[137…

	id	mass (g)	reclat	reclong
count	45716.000000	4.558500e+04	38401.000000	38401.000000
mean	26889.735104	1.327808e+04	-39.122580	61.074319
std	16860.683030	5.749889e+05	46.378511	80.647298
min	1.000000	0.000000e+00	-87.366670	-165.433330
25%	12688.750000	7.200000e+00	-76.714240	0.000000
50%	24261.500000	3.260000e+01	-71.500000	35.666670
75%	40656.750000	2.026000e+02	0.000000	157.166670
max	57458.000000	6.000000e+07	81.166670	354.473330

In [138…
```
meteorites.describe(include = 'all')
```

Out[138...

	name	id	nametype	recclass	mass (g)	fall	year	
count	45716	45716.000000	45716	45716	4.558500e+04	45716	45425	3840
unique	45716	NaN	2	466	NaN	2	266	
top	Aachen	NaN	Valid	L6	NaN	Found	01/01/2003 12:00:00 AM	
freq	1	NaN	45641	8285	NaN	44609	3323	
mean	NaN	26889.735104	NaN	NaN	1.327808e+04	NaN	NaN	-3
std	NaN	16860.683030	NaN	NaN	5.749889e+05	NaN	NaN	4
min	NaN	1.000000	NaN	NaN	0.000000e+00	NaN	NaN	-8
25%	NaN	12688.750000	NaN	NaN	7.200000e+00	NaN	NaN	-7
50%	NaN	24261.500000	NaN	NaN	3.260000e+01	NaN	NaN	-7
75%	NaN	40656.750000	NaN	NaN	2.026000e+02	NaN	NaN	
max	NaN	57458.000000	NaN	NaN	6.000000e+07	NaN	NaN	8

Exercise (Part 1)

Using the 2019_Yellow_Taxi_Trip_Data.csv dataset, accomplish the following items and submit a PDF of the notebook:e).

1. Create a DataFrame by reading in the 2019_Yellow_Taxi_Trip_Data.csv file. Examine the first 5 rows.

In [142...

```
taxi = pd.read_csv('2019_Yellow_Taxi_Trip_Data.csv')
taxi.head()
```

Out[142...

	vendorid	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance
0	2	2019-10-23T16:39:42.000	2019-10-23T17:14:10.000	1	7.93
1	1	2019-10-23T16:32:08.000	2019-10-23T16:45:26.000	1	2.00
2	2	2019-10-23T16:08:44.000	2019-10-23T16:21:11.000	1	1.36
3	2	2019-10-23T16:22:44.000	2019-10-23T16:43:26.000	1	1.00
4	2	2019-10-23T16:45:11.000	2019-10-23T16:58:49.000	1	1.96

2. Find the dimensions (number of rows and number of columns) in the data.

In [176…
```
rows, columns = taxi.shape
print(f'Rows: {rows}')
print(f'Columns: {columns}')
```

```
Rows: 10000
Columns: 18
```

3. Using the data in the 2019_Yellow_Taxi_Trip_Data.csv file, calculate summary statistics for the fare_amount, tip_amount, tolls_amount, and total_amount columns.

In [149…
```
taxi.describe()[['fare_amount', 'tip_amount', 'tolls_amount', 'total_amount']]
```

Out[149…

	fare_amount	tip_amount	tolls_amount	total_amount
count	10000.000000	10000.000000	10000.000000	10000.000000
mean	15.106313	2.634494	0.623447	22.564659
std	13.954762	3.409800	6.437507	19.209255
min	-52.000000	0.000000	-6.120000	-65.920000
25%	7.000000	0.000000	0.000000	12.375000
50%	10.000000	2.000000	0.000000	16.300000
75%	16.000000	3.250000	0.000000	22.880000
max	176.000000	43.000000	612.000000	671.800000

4. Isolate the fare_amount, tip_amount, tolls_amount, and total_amount for the longest trip by distance (trip_distance).

In [179…
```
taxi.iloc[taxi['trip_distance'].idxmax()][['fare_amount', 'tip_amount', 'tolls_amou
```

Out[179…
```
fare_amount      176.0
tip_amount       18.29
tolls_amount      6.12
total_amount    201.21
Name: 8338, dtype: object
```

Reflection: In this acitivity I was able to learn something new about python pandas, such as describe, shape, max, median, info, and using boolean in pandas. I wasn't able to explore those python pandas function before in our EDA because I was more focused on what would be useful in implementing, and for the completion of our activities and projects.