```
In [36]:   import pandas as pd

           meteorites = pd.read_csv('Meteorite_Landings.csv',nrows=5)
           meteorites
```

Out[36]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong
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 [5]:   meteorites.name
```

```
Out[5]:   0       Aachen
          1       Aarhus
          2         Abee
          3     Acapulco
          4      Achiras
          Name: name, dtype: object
```

```
In [12]:   meteorites['name']
```

```
Out[12]:   0       Aachen
           1       Aarhus
           2         Abee
           3     Acapulco
           4      Achiras
           Name: name, dtype: object
```

```
In [11]:   meteorites.columns
```

```
Out[11]:   Index(['name', 'id', 'nametype', 'recclass', 'mass (g)', 'fall', 'year',
                  'reclat', 'reclong', 'GeoLocation'],
                 dtype='object')
```

```
In [14]:   meteorites.index
```

```
Out[14]:   RangeIndex(start=0, stop=5, step=1)
```

```python
In [17]:  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
```

```python
In [22]:  payload[0]
```

```python
Out[22]:  {'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'}}
```

```python
In [23]:  df = pd.DataFrame(payload)
          df.head(3)
```

Out[23]:

	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	

```python
In [149…  meteorites = pd.read_csv('Meteorite_Landings.csv')
```

```python
In [29]:  meteorites.shape
```

```
Out[29]:  (45716, 10)
```

```
In [30]: meteorites.columns
```

Out[30]: Index(['name', 'id', 'nametype', 'recclass', 'mass (g)', 'fall', 'year',
               'reclat', 'reclong', 'GeoLocation'],
              dtype='object')

```
In [33]: meteorites.dtypes
```

Out[33]: name   object
         id   int64
         nametype   object
         recclass   object
         mass (g)   float64
         fall   object
         year   object
         reclat   float64
         reclong   float64
         GeoLocation   object
         dtype: object

```
In [43]: meteorites.head(10)
```

Out[43]:

	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 [35]: `meteorites.tail(5)`

Out[35]:

	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	ε
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 [47]: `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 [52]: `meteorites[['name','mass (g)']]`

Out[52]:

	name	mass (g)
0	Aachen	21.0
1	Aarhus	720.0
2	Abee	107000.0
3	Acapulco	1914.0
4	Achiras	780.0
...	...	...
45711	Zillah 002	172.0
45712	Zinder	46.0
45713	Zlin	3.3
45714	Zubkovsky	2167.0
45715	Zulu Queen	200.0

45716 rows × 2 columns

In [53]: `meteorites[100:104]`

Out[53]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclon
100	Benton	5026	Valid	LL6	2840.0	Fell	01/01/1949 12:00:00 AM	45.95000	-67.5500
101	Berduc	48975	Valid	L6	270.0	Fell	01/01/2008 12:00:00 AM	-31.91000	-58.3283
102	Béréba	5028	Valid	Eucrite-mmict	18000.0	Fell	01/01/1924 12:00:00 AM	11.65000	-3.6500
103	Berlanguillas	5029	Valid	L6	1440.0	Fell	01/01/1811 12:00:00 AM	41.68333	-3.8000

In [58]: `meteorites.iloc[100:104, [0,3,4,6]]`

	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 [60]:
```python
meteorites.loc[100:104, 'mass (g)':'year']
```

Out[60]:

	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 [69]:
```python
meteorites.iloc[-1, [-1]]
```

Out[69]:
```
GeoLocation    (33.98333, -115.68333)
Name: 45715, dtype: object
```

In [71]:
```python
(meteorites['mass (g)'] > 50) & (meteorites.fall == 'Found')
```

Out[71]:
```
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 [83]:
```python
meteorites[(meteorites['mass (g)'] > 1e6) & (meteorites.fall == 'Fell')]
```

`Out[83]:`

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong
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 [81]:`
```python
meteorites.query("`mass (g)` > 1e6 & fall == 'Fell'")
```

`Out[81]:`

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong
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 [84]:`
```python
meteorites.fall.value_counts()
```

`Out[84]:`
```
fall
Found    44609
Fell      1107
Name: count, dtype: int64
```

`In [90]:`
```python
meteorites.value_counts(subset=['nametype','fall'], normalize = True)
```

```
Out[90]:    nametype  fall
            Valid     Found    0.974145
                      Fell     0.024215
            Relict    Found    0.001641
            Name: proportion, dtype: float64
```

```
In [103…    meteorites['mass (g)'].mean()
```

```
Out[103…    13278.078548601512
```

```
In [110…    meteorites['mass (g)'].quantile(0.75)
```

```
Out[110…    202.6
```

```
In [100…    meteorites['mass (g)'].median()
```

```
Out[100…    32.6
```

```
In [101…    meteorites['mass (g)'].max()
```

```
Out[101…    60000000.0
```

```
In [102…    meteorites.loc[meteorites['mass (g)'].idxmax()]
```

```
Out[102…    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 [111…    meteorites.recclass.nunique()
```

```
Out[111…    466
```

```
In [115…    meteorites.recclass.unique()[:14]
```

```
Out[115…    array(['L5', 'H6', 'EH4', 'Acapulcoite', 'L6', 'LL3-6', 'H5', 'L',
                   'Diogenite-pm', 'Unknown', 'H4', 'H', 'Iron, IVA', 'CR2-an'],
                  dtype=object)
```

```
In [112…    meteorites.name.nunique()
```

```
Out[112…    45716
```

```
In [147…    meteorites.describe(include='all')
```

|	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	-39
std	NaN	16860.683030	NaN	NaN	5.749889e+05	NaN	NaN	46
min	NaN	1.000000	NaN	NaN	0.000000e+00	NaN	NaN	-87
25%	NaN	12688.750000	NaN	NaN	7.200000e+00	NaN	NaN	-76
50%	NaN	24261.500000	NaN	NaN	3.260000e+01	NaN	NaN	-71
75%	NaN	40656.750000	NaN	NaN	2.026000e+02	NaN	NaN	0
max	NaN	57458.000000	NaN	NaN	6.000000e+07	NaN	NaN	81

```python
#Excercise (Part 1)

#1.)
df = pd.read_csv('2019_Yellow_Taxi_Trip_Data.csv')
df
```

	vendorid	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distan
0	2	2019-10-23T16:39:42.000	2019-10-23T17:14:10.000	1	7.
1	1	2019-10-23T16:32:08.000	2019-10-23T16:45:26.000	1	2.
2	2	2019-10-23T16:08:44.000	2019-10-23T16:21:11.000	1	1.
3	2	2019-10-23T16:22:44.000	2019-10-23T16:43:26.000	1	1.
4	2	2019-10-23T16:45:11.000	2019-10-23T16:58:49.000	1	1.
...	...	...	...	...	
9995	1	2019-10-23T17:39:59.000	2019-10-23T17:49:26.000	2	1.
9996	1	2019-10-23T17:53:02.000	2019-10-23T18:00:45.000	1	1.
9997	1	2019-10-23T17:07:16.000	2019-10-23T17:11:35.000	1	0.
9998	1	2019-10-23T17:38:26.000	2019-10-23T17:49:28.000	2	2.
9999	1	2019-10-23T17:22:14.000	2019-10-23T17:52:09.000	1	3.

10000 rows × 18 columns

```
#2)

df.shape
```
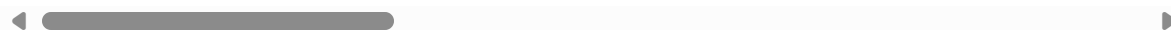
(10000, 18)

```
#3)

df1 = df.iloc[:, [10,13,14,16]]
df1.describe()
```

	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

In [166...

```python
#4)

df1.loc[df1['trip_distance'].idxmax()]
```

Out[166...

```
trip_distance      38.11
fare_amount       176.00
tip_amount         18.29
tolls_amount        6.12
total_amount      201.21
Name: 8338, dtype: float64
```

Observation: I experienced difficulties in Practicing the codes discussed, but as I started to get the flow, I hadn't had to go back and see the codes done for reference, I remembered some of the codes although not all.

In [2]:
```python
import pandas as pd

newdf = pd.read_csv('2019_Yellow_Taxi_Trip_Data.csv')
```

In [8]:
```python
mask = newdf.columns.str.contains('id$|store_and_fwd_flag', regex = True)
columns_to_drop = newdf.columns[mask]
columns_to_drop
```

Out[8]:
```
Index(['vendorid', 'ratecodeid', 'store_and_fwd_flag', 'pulocationid',
       'dolocationid'],
      dtype='object')
```

In [9]:
```python
taxis = newdf.drop(columns=columns_to_drop)
taxis.head()
```

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

◄ ▬▬▬▬▬▬▬▬▬▬▬▬▬ ►

```python
taxis = taxis.rename(
    columns={
        'tpep_pickup_datetime': 'pickup_time',
        'tpep_dropoff_datetime': 'dropoff_time'
    }
)
taxis.columns
```

```
Index(['pickup_time', 'dropoff_time', 'passenger_count', 'trip_distance',
       'payment_type', 'fare_amount', 'extra', 'mta_tax', 'tip_amount',
       'tolls_amount', 'improvement_surcharge', 'total_amount',
       'congestion_surcharge'],
      dtype='object')
```

```python
taxis[['pickup_time','dropoff_time']] = taxis[['pickup_time','dropoff_time']].apply
taxis.dtypes
```

```
pickup_time              datetime64[ns]
dropoff_time             datetime64[ns]
passenger_count                   int64
trip_distance                   float64
payment_type                      int64
fare_amount                     float64
extra                           float64
mta_tax                         float64
tip_amount                      float64
tolls_amount                    float64
improvement_surcharge           float64
total_amount                    float64
congestion_surcharge            float64
dtype: object
```

```python
taxis = taxis.assign(
    elapsed_time=lambda x: x.dropoff_time - x.pickup_time,
    cost_before_tip=lambda x: x.total_amount - x.tip_amount,
    tip_pct=lambda x: x.tip_amount / x.cost_before_tip,
    fees=lambda x: x.cost_before_tip - x.fare_amount,
    avg_speed=lambda x: x.trip_distance.div(x.elapsed_time.dt.total_seconds()/60/60
```

```
                                    )
        )
```

In [19]: `taxis.dtypes`

Out[19]:
```
pickup_time               datetime64[ns]
dropoff_time              datetime64[ns]
passenger_count                    int64
trip_distance                    float64
payment_type                       int64
fare_amount                      float64
extra                            float64
mta_tax                          float64
tip_amount                       float64
tolls_amount                     float64
improvement_surcharge            float64
total_amount                     float64
congestion_surcharge             float64
elapsed_time              timedelta64[ns]
cost_before_tip                  float64
tip_pct                          float64
fees                             float64
avg_speed                        float64
dtype: object
```

In [22]: `taxis.sort_values(['passenger_count','pickup_time'],ascending=[False,True]).head()`

Out[22]:

	pickup_time	dropoff_time	passenger_count	trip_distance	payment_type	fare_amou
5997	2019-10-23 15:55:19	2019-10-23 16:08:25	6	1.58	2	10
443	2019-10-23 15:56:59	2019-10-23 16:04:33	6	1.46	2	7
8722	2019-10-23 15:57:33	2019-10-23 16:03:34	6	0.62	1	5
4198	2019-10-23 15:57:38	2019-10-23 16:05:07	6	1.18	1	7
8238	2019-10-23 15:58:31	2019-10-23 16:29:29	6	3.23	2	19

In [24]: `taxis.head()`

	pickup_time	dropoff_time	passenger_count	trip_distance	payment_type	fare_amount
0	2019-10-23 16:39:42	2019-10-23 17:14:10	1	7.93	1	29.5
1	2019-10-23 16:32:08	2019-10-23 16:45:26	1	2.00	1	10.5
2	2019-10-23 16:08:44	2019-10-23 16:21:11	1	1.36	1	9.5
3	2019-10-23 16:22:44	2019-10-23 16:43:26	1	1.00	1	13.0
4	2019-10-23 16:45:11	2019-10-23 16:58:49	1	1.96	1	10.5

```python
taxis.nlargest(3,'elapsed_time')
```

	pickup_time	dropoff_time	passenger_count	trip_distance	payment_type	fare_amou
7576	2019-10-23 16:52:51	2019-10-24 16:51:44	1	3.75	1	1
6902	2019-10-23 16:51:42	2019-10-24 16:50:22	1	11.19	2	39
4975	2019-10-23 16:18:51	2019-10-24 16:17:30	1	0.70	2	7

```python
taxis.nlargest(3, 'trip_distance')
```

	pickup_time	dropoff_time	passenger_count	trip_distance	payment_type	fare_amou
8338	2019-10-23 16:50:53	2019-10-24 15:32:55	1	38.11	1	176
9965	2019-10-23 17:34:29	2019-10-23 18:48:00	1	37.86	2	52
1656	2019-10-23 16:04:45	2019-10-23 19:11:40	3	37.57	1	52

```python
#Seatwork 2
```

```python
met = pd.read_csv('Meteorite_Landings.csv')
met.head()
```

Out[37]:

	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

◀ ━━━━━━━━━━━━━━━━━━━━━━━━━━ ▶

In [38]:
```python
met = met.rename(
    columns ={
    'mass (g)': 'mass'
}
)
met.head()
```

Out[38]:

	name	id	nametype	recclass	mass	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

◀ ━━━━━━━━━━━━━━━━━━━━━━━━━━ ▶

In [39]:
```python
met = met.drop(columns=['reclat','reclong'])
met.head()
```

	name	id	nametype	recclass	mass	fall	year	GeoLocation
0	Aachen	1	Valid	L5	21.0	Fell	01/01/1880 12:00:00 AM	(50.775, 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, -113.0)
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	01/01/1976 12:00:00 AM	(16.88333, -99.9)
4	Achiras	370	Valid	L6	780.0	Fell	01/01/1902 12:00:00 AM	(-33.16667, -64.95)

```python
In [44]: met = met.sort_values('mass',ascending=False)
met.head()
```

	name	id	nametype	recclass	mass	fall	year	GeoLocation
16392	Hoba	11890	Valid	Iron, IVB	60000000.0	Found	01/01/1920 12:00:00 AM	(-19.58333, 17.91667)
5373	Cape York	5262	Valid	Iron, IIIAB	58200000.0	Found	01/01/1818 12:00:00 AM	(76.13333, -64.93333)
5365	Campo del Cielo	5247	Valid	Iron, IAB-MG	50000000.0	Found	12/22/1575 12:00:00 AM	(-27.46667, -60.58333)
5370	Canyon Diablo	5257	Valid	Iron, IAB-MG	30000000.0	Found	01/01/1891 12:00:00 AM	(35.05, -111.03333)
3455	Armanty	2335	Valid	Iron, IIIE	28000000.0	Found	01/01/1898 12:00:00 AM	(47.0, 88.0)

In [ ]: