In [2]:

```
import pandas as pd

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

Out[2]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong	GeoLocation
0	Aachen	1	Valid	L5	21	Fell	01/01/1880 12:00:00 AM	50.77500	6.08333	(50.775, 6.08333)
1	Aarhus	2	Valid	H6	720	Fell	01/01/1951 12:00:00 AM	56.18333	10.23333	(56.18333, 10.23333)
2	Abee	6	Valid	EH4	107000	Fell	01/01/1952 12:00:00 AM	54.21667	-113.00000	(54.21667, -113.0)
3	Acapulco	10	Valid	Acapulcoite	1914	Fell	01/01/1976 12:00:00 AM	16.88333	-99.90000	(16.88333, -99.9)
4	Achiras	370	Valid	L6	780	Fell	01/01/1902 12:00:00 AM	-33.16667	-64.95000	(-33.16667, -64.95)

In [3]:

```
meteorites.name
```

Out[3]:

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

In [5]:

```
meteorites["name"]
```

Out[5]:

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

In [6]:

```
meteorites.columns
```

Out[6]:

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

In [8]:

```
meteorites.index
```

Out[8]:

```
RangeIndex(start=0, stop=5, step=1)
```

In [13]:

```
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 unsuccessful and returned code: {response.status_code}.")
    payload = None
```

In [17]:

```
import pandas as pd

df = pd.DataFrame(payload)
df.head(3)
```

Out[17]:

	name	id	nametype	recclass	mass	fall	year	reclat	reclong	geolocation	:@computed_region_cbhk_fwbd	:@c
0	Aachen	1	Valid	L5	21	Fell	1880-01-01T00:00:00.000	50.775000	6.083330	{'latitude': '50.775', 'longitude': '6.08333'}	NaN	
1	Aarhus	2	Valid	H6	720	Fell	1951-01-01T00:00:00.000	56.183330	10.233330	{'latitude': '56.18333', 'longitude': '10.23333'}	NaN	
2	Abee	6	Valid	EH4	107000	Fell	1952-01-01T00:00:00.000	54.216670	-113.000000	{'latitude': '54.21667', 'longitude': '-113.0'}	NaN	

In [23]:

```
meteorites.shape
```

Out[23]:

```
(45716, 10)
```

In [24]:

```
meteorites.columns
```

Out[24]:

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

In [25]:

```
meteorites.dtypes
```

Out[25]:

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

In [27]:

```
meteorites.head()
```

Out[27]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong	GeoLocation
0	Aachen	1	Valid	L5	21.0	Fell	01/01/1880 12:00:00 AM	50.77500	6.08333	(50.775, 6.08333)
1	Aarhus	2	Valid	H6	720.0	Fell	01/01/1951 12:00:00 AM	56.18333	10.23333	(56.18333, 10.23333)
2	Abee	6	Valid	EH4	107000.0	Fell	01/01/1952 12:00:00 AM	54.21667	-113.00000	(54.21667, -113.0)
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	01/01/1976 12:00:00 AM	16.88333	-99.90000	(16.88333, -99.9)
4	Achiras	370	Valid	L6	780.0	Fell	01/01/1902 12:00:00 AM	-33.16667	-64.95000	(-33.16667, -64.95)

In [28]:

```
meteorites.tail()
```

Out[28]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong	GeoLocation
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	01/01/1990 12:00:00 AM	29.03700	17.01850	(29.037, 17.0185)
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	01/01/1999 12:00:00 AM	13.78333	8.96667	(13.78333, 8.96667)
45713	Zlin	30410	Valid	H4	3.3	Found	01/01/1939 12:00:00 AM	49.25000	17.66667	(49.25, 17.66667)
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	01/01/2003 12:00:00 AM	49.78917	41.50460	(49.78917, 41.5046)
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	01/01/1976 12:00:00 AM	33.98333	-115.68333	(33.98333, -115.68333)

In [35]:

```
# View the first 10 entries of the dataset
meteorites.head(10)
```

Out[35]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong	GeoLocation
0	Aachen	1	Valid	L5	21.0	Fell	01/01/1880 12:00:00 AM	50.77500	6.08333	(50.775, 6.08333)
1	Aarhus	2	Valid	H6	720.0	Fell	01/01/1951 12:00:00 AM	56.18333	10.23333	(56.18333, 10.23333)
2	Abee	6	Valid	EH4	107000.0	Fell	01/01/1952 12:00:00 AM	54.21667	-113.00000	(54.21667, -113.0)
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	01/01/1976 12:00:00 AM	16.88333	-99.90000	(16.88333, -99.9)
4	Achiras	370	Valid	L6	780.0	Fell	01/01/1902 12:00:00 AM	-33.16667	-64.95000	(-33.16667, -64.95)
5	Adhi Kot	379	Valid	EH4	4239.0	Fell	01/01/1919 12:00:00 AM	32.10000	71.80000	(32.1, 71.8)
6	Adzhi-Bogdo (stone)	390	Valid	LL3-6	910.0	Fell	01/01/1949 12:00:00 AM	44.83333	95.16667	(44.83333, 95.16667)
7	Agen	392	Valid	H5	30000.0	Fell	01/01/1814 12:00:00 AM	44.21667	0.61667	(44.21667, 0.61667)
8	Aguada	398	Valid	L6	1620.0	Fell	01/01/1930 12:00:00 AM	-31.60000	-65.23333	(-31.6, -65.23333)
9	Aguila Blanca	417	Valid	L	1440.0	Fell	01/01/1920 12:00:00 AM	-30.86667	-64.55000	(-30.86667, -64.55)

In [32]:

```
# View the last 5 rows of the dataset
meteorites.tail(5)
```

Out[32]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong	GeoLocation
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	01/01/1990 12:00:00 AM	29.03700	17.01850	(29.037, 17.0185)
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	01/01/1999 12:00:00 AM	13.78333	8.96667	(13.78333, 8.96667)
45713	Zlin	30410	Valid	H4	3.3	Found	01/01/1939 12:00:00 AM	49.25000	17.66667	(49.25, 17.66667)
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	01/01/2003 12:00:00 AM	49.78917	41.50460	(49.78917, 41.5046)
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	01/01/1976 12:00:00 AM	33.98333	-115.68333	(33.98333, -115.68333)

In [38]:

```
# Get some information about the data
meteorites.info() # object = strings
                  # used for viewing missing data
```

```
<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 [43]:

```
# Select multiple rows
meteorites[["name","id"]]
```

Out[43]:

	name	id
0	Aachen	1
1	Aarhus	2
2	Abee	6
3	Acapulco	10
4	Achiras	370
...	...	...
45711	Zillah 002	31356
45712	Zinder	30409
45713	Zlin	30410
45714	Zubkovsky	31357
45715	Zulu Queen	30414

45716 rows × 2 columns

In [44]:

```
meteorites[100:104]
```

Out[44]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong	GeoLocation
100	Benton	5026	Valid	LL6	2840.0	Fell	01/01/1949 12:00:00 AM	45.95000	-67.55000	(45.95, -67.55)
101	Berduc	48975	Valid	L6	270.0	Fell	01/01/2008 12:00:00 AM	-31.91000	-58.32833	(-31.91, -58.32833)
102	Béréba	5028	Valid	Eucrite-mmict	18000.0	Fell	01/01/1924 12:00:00 AM	11.65000	-3.65000	(11.65, -3.65)
103	Berlanguillas	5029	Valid	L6	1440.0	Fell	01/01/1811 12:00:00 AM	41.68333	-3.80000	(41.68333, -3.8)

In [45]:

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

Out[45]:

	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 [48]:

```
# loc is used if we want to acces definite comulmn names
meteorites.loc[100:104, 'mass (g)':'year']
```

Out[48]:

	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 [51]:

```
# Access the last row last column
meteorites.iloc[-1, -1]
```

Out[51]:

```
'(33.98333, -115.68333)'
```

In [ ]:

```
# Filtering with Boolean Masks
```

In [53]:

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

Out[53]:

```
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 [54]:

```
meteorites[(meteorites['mass (g)'] > 1e6) & (meteorites.fall == 'Fell')]
```

Out[54]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong	GeoLocation
29	Allende	2278	Valid	CV3	2000000.0	Fell	01/01/1969 12:00:00 AM	26.96667	-105.31667	(26.96667, -105.31667)
419	Jilin	12171	Valid	H5	4000000.0	Fell	01/01/1976 12:00:00 AM	44.05000	126.16667	(44.05, 126.16667)
506	Kunya-Urgench	12379	Valid	H5	1100000.0	Fell	01/01/1998 12:00:00 AM	42.25000	59.20000	(42.25, 59.2)
707	Norton County	17922	Valid	Aubrite	1100000.0	Fell	01/01/1948 12:00:00 AM	39.68333	-99.86667	(39.68333, -99.86667)
920	Sikhote-Alin	23593	Valid	Iron, IIAB	23000000.0	Fell	01/01/1947 12:00:00 AM	46.16000	134.65333	(46.16, 134.65333)

In [56]:

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

Out[56]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong	GeoLocation
29	Allende	2278	Valid	CV3	2000000.0	Fell	01/01/1969 12:00:00 AM	26.96667	-105.31667	(26.96667, -105.31667)
419	Jilin	12171	Valid	H5	4000000.0	Fell	01/01/1976 12:00:00 AM	44.05000	126.16667	(44.05, 126.16667)
506	Kunya-Urgench	12379	Valid	H5	1100000.0	Fell	01/01/1998 12:00:00 AM	42.25000	59.20000	(42.25, 59.2)
707	Norton County	17922	Valid	Aubrite	1100000.0	Fell	01/01/1948 12:00:00 AM	39.68333	-99.86667	(39.68333, -99.86667)
920	Sikhote-Alin	23593	Valid	Iron, IIAB	23000000.0	Fell	01/01/1947 12:00:00 AM	46.16000	134.65333	(46.16, 134.65333)

In [ ]:

```
# Calculating Statistics
```

In [57]:

```
meteorites.fall.value_counts() # Counts Fall not null values
```

Out[57]:

```
fall
Found    44609
Fell      1107
Name: count, dtype: int64
```

In [61]:

```
meteorites.value_counts(subset = ['nametype', 'fall'], normalize = True) # Count unique values
```

Out[61]:

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

In [67]:

```
# meteorites['mass (g)'].mean()
float(meteorites['mass (g)'].mean())
```

Out[67]:

```
13278.078548601512
```

In [66]:

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

Out[66]:

```
numpy.float64
```

In [62]:

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

Out[62]:

```
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 [68]:

```
meteorites['mass (g)'].median()
```

Out[68]:

```
32.6
```

In [73]:

```
meteorites['mass (g)'].max()
```

Out[73]:

```
60000000.0
```

In [71]:

```
meteorites.loc[meteorites['mass (g)'].idxmax()] # Locate the index of the max value
```

Out[71]:

```
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 [76]:

```
meteorites.recclass.nunique() # There are repeating
```

Out[76]:

```
466
```

In [75]:

```
meteorites.recclass.unique()[:14]
```

Out[75]:

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

In [79]:

```
# meteorites.describe() # numerical values only
meteorites.describe(include = 'all')
```

Out[79]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong	GeoLocation
count	45716	45716.000000	45716	45716	4.558500e+04	45716	45425	38401.000000	38401.000000	38401
unique	45716	NaN	2	466	NaN	2	266	NaN	NaN	17100
top	Aachen	NaN	Valid	L6	NaN	Found	01/01/2003 12:00:00 AM	NaN	NaN	(0.0, 0.0)
freq	1	NaN	45641	8285	NaN	44609	3323	NaN	NaN	6214
mean	NaN	26889.735104	NaN	NaN	1.327808e+04	NaN	NaN	-39.122580	61.074319	NaN
std	NaN	16860.683030	NaN	NaN	5.749889e+05	NaN	NaN	46.378511	80.647298	NaN
min	NaN	1.000000	NaN	NaN	0.000000e+00	NaN	NaN	-87.366670	-165.433330	NaN
25%	NaN	12688.750000	NaN	NaN	7.200000e+00	NaN	NaN	-76.714240	0.000000	NaN
50%	NaN	24261.500000	NaN	NaN	3.260000e+01	NaN	NaN	-71.500000	35.666670	NaN
75%	NaN	40656.750000	NaN	NaN	2.026000e+02	NaN	NaN	0.000000	157.166670	NaN
max	NaN	57458.000000	NaN	NaN	6.000000e+07	NaN	NaN	81.166670	354.473330	NaN

Execrise (Part 1)

In [86]:

```
# 1. Create a DataFrame by reading in the 2019_Yellow_Taxi_Trip_Data.csv file. Examine the first 5 rows.
import pandas as pd

yellow_taxi = pd.read_csv('2019_Yellow_Taxi_Trip_Data.csv')

yellow_taxi.head(5)
```

Out[86]:

	vendorid	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	ratecodeid	store_and_fwd_flag	pulocationid	d
0	2	2019-10-23T16:39:42.000	2019-10-23T17:14:10.000	1	7.93	1	N	138	
1	1	2019-10-23T16:32:08.000	2019-10-23T16:45:26.000	1	2.00	1	N	11	
2	2	2019-10-23T16:08:44.000	2019-10-23T16:21:11.000	1	1.36	1	N	163	
3	2	2019-10-23T16:22:44.000	2019-10-23T16:43:26.000	1	1.00	1	N	170	
4	2	2019-10-23T16:45:11.000	2019-10-23T16:58:49.000	1	1.96	1	N	163	

In [87]:

```
# 2. Find the dimensions (number of rows and number of columns) in the data.
yellow_taxi.shape
```

Out[87]:

```
(10000, 18)
```

In [101…

```
# 3. Using the data in the 2019_Yellow_Taxi_Trip_Data.csv file, calculate summary statistics for the fare_amount, tip_amount,

summary_stats = df[['fare_amount', 'tip_amount', 'tolls_amount', 'total_amount']].describe()
print("Summary Statistics:")
summary_stats
```

```
Summary Statistics:
```

Out[101…

	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 [103…

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

longest_trip = yellow_taxi.loc[yellow_taxi['trip_distance'].idxmax()]
print("Longest Trip by Distance:")
longest_trip[['fare_amount', 'tip_amount', 'tolls_amount', 'total_amount']]
```

```
Longest Trip by Distance:
```

Out[103…

```
fare_amount      176.0
tip_amount       18.29
tolls_amount      6.12
total_amount    201.21
Name: 8338, dtype: object
```

Reflection:

In this activity, I have learned more about pandas functions. Similarly, this served as a refesher to previous VDA class about pandas. Moreso, I became knowledgeable of more functions important to data science I do not know before. Although I felt a bit overwhelmed with how vast inner functions pandas, especially the "complex" form of having several brackets and "dot" method calls.