In [19]:
```
import pandas as pd

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

Out[19]:

	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	

Series

In [15]:
```
meteorites.name
```

Out[15]:
```
0      Aachen
1      Aarhus
2        Abee
3    Acapulco
4     Achiras
Name: name, dtype: object
```

In [16]:
```
meteorites['name']
```

Out[16]:
```
0      Aachen
1      Aarhus
2        Abee
3    Acapulco
4     Achiras
Name: name, dtype: object
```

Columns

In [17]:
```
meteorites.columns
```

Out[17]:
```
Index(['name', 'id', 'nametype', 'recclass', 'mass (g)', 'fall', 'year',
       'reclat', 'reclong', 'GeoLocation'],
      dtype='object')
```

Index

In [18]:
```
meteorites.index
```

Out[18]:
```
RangeIndex(start=0, stop=5, step=1)
```

In [29]:
```
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 succesful and returned code: {response.status_code}')
    payload = None
```

In [ ]:
```
payload
```

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

In [34]:
```
import pandas as pd

meteorites = pd.read_csv('Meteorite_Landings.csv')
```

How many rows and columns are there?

In [35]:
```
meteorites.shape
```

Out[35]:
```
(45716, 10)
```

What are the column names?

In [37]:
```
meteorites.columns
```

Out[37]:
```
Index(['name', 'id', 'nametype', 'recclass', 'mass (g)', 'fall', 'year',
       'reclat', 'reclong', 'GeoLocation'],
      dtype='object')
```

What are the dtypes?

In [41]:
```
meteorites.dtypes
```

Out[41]:
```
name            object
id               int64
nametype        object
recclass        object
mass (g)       float64
fall            object
year            object
reclat         float64
reclong        float64
GeoLocation     object
dtype: object
```

First 10 and last 10 of the dataset "Meteorites.csv"

In [42]:
```
meteorites.head(10) # first ten
```

Out[42]:

	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 [43]:
```
meteorites.tail(5) #last five
```

Out[43]:

	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

Get some information about the DataFrame

In [44]:
```
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 [ ]:
```
#Extracting Subsets
```

In [45]:
```
meteorites
```

Out[45]:

	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 [49]:
```
meteorites[['name','fall']]
```

Out[49]:

	name	fall
0	Aachen	Fell
1	Aarhus	Fell
2	Abee	Fell
3	Acapulco	Fell
4	Achiras	Fell
...	...	...
45711	Zillah 002	Found
45712	Zinder	Found
45713	Zlin	Found
45714	Zubkovsky	Found
45715	Zulu Queen	Found

45716 rows × 2 columns

In [51]:
```
#selecting rows
meteorites[100:104]
```

Out[51]:

	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 [57]:
```
#Indexing
meteorites.iloc[:,[0,3,4,6]]
```

Out[57]:

	name	recclass	mass (g)	year
0	Aachen	L5	21.0	01/01/1880 12:00:00 AM
1	Aarhus	H6	720.0	01/01/1951 12:00:00 AM
2	Abee	EH4	107000.0	01/01/1952 12:00:00 AM
3	Acapulco	Acapulcoite	1914.0	01/01/1976 12:00:00 AM
4	Achiras	L6	780.0	01/01/1902 12:00:00 AM
...	...	...	...	...
45711	Zillah 002	Eucrite	172.0	01/01/1990 12:00:00 AM
45712	Zinder	Pallasite, ungrouped	46.0	01/01/1999 12:00:00 AM
45713	Zlin	H4	3.3	01/01/1939 12:00:00 AM
45714	Zubkovsky	L6	2167.0	01/01/2003 12:00:00 AM
45715	Zulu Queen	L3.7	200.0	01/01/1976 12:00:00 AM

45716 rows × 4 columns

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

In [61]:
```
meteorites.loc[100:104,'mass (g)']
```

Out[61]:
```
100     2840.0
101      270.0
102    18000.0
103     1440.0
104      960.0
Name: mass (g), dtype: float64
```

In [65]:
```
#last column and last row

meteorites.iloc[-1::]
```

Out[65]:

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	01/01/1976 12:00:00 AM	33.98333	-115.68333

In [71]:
```
meteorites.iloc[-1,-1]
```

Out[71]:
```
'(33.98333, -115.68333)'
```

In [72]:
```
(meteorites['mass (g)'] > 50) & (meteorites.fall == 'Found')
```

Out[72]:
```
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 [73]:
```
meteorites[(meteorites['mass (g)'] > 50) & (meteorites.fall == 'Found')]
```

Out[73]:

	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 [75]:
```
meteorites[(meteorites['mass (g)'] > 1e6) & (meteorites.fall == 'Fell')]
```

Out[75]:

	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 [80]:
```
meteorites.query("`mass (g)` > 1e6 and fall == 'Fell' ")
```

Out[80]:

	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 [86]:
```
#how many of the meteorite were found versus observed falling?

meteorites.fall.value_counts()
```

Out[86]:
```
fall
Found    44609
Fell      1107
Name: count, dtype: int64
```

In [91]:
```
meteorites.value_counts(subset = ['nametype','fall'], normalize = True)
```

Out[91]:
```
nametype  fall
Valid     Found    0.974145
          Fell     0.024215
Relict    Found    0.001641
Name: proportion, dtype: float64
```

In [92]:
```
meteorites.value_counts(subset = ['nametype','fall'])
```

Out[92]:
```
nametype  fall
Valid     Found    44534
          Fell      1107
Relict    Found       75
Name: count, dtype: int64
```

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

Out[101…
```
numpy.float64
```

In [103…
```
#what was the mass of the average meteorite?

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

```
13278.078548601512
```

In [99]:
```
meteorites['mass (g)'].quantile([0.01,0.05,0.5,0.95,0.99])
```

Out[99]:
```
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 [104…
```
meteorites['mass (g)'].median()
```

Out[104…
```
32.6
```

In [109…
```
#what was the mass of the heaviest meteorite?
meteorites['mass (g)'].max()
```

Out[109…
```
60000000.0
```

In [111…
```
meteorites.loc[meteorites['mass (g)'].idxmax()]
```

Out[111…
```
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 [112…
```
meteorites.recclass.nunique()
```

Out[112…
```
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 [114…
```
meteorites.name.unique
```

Out[114…
```
<bound method Series.unique of 0            Aachen
1             Aarhus
2               Abee
3            Acapulco
4             Achiras
              ...
45711    Zillah 002
45712         Zinder
45713           Zlin
45714      Zubkovsky
45715    Zulu Queen
Name: name, Length: 45716, dtype: object>
```

In [117…
```
meteorites.describe()
```

Out[117…

	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 [118…
```
meteorites.describe(include='all')
```

Out[118…

	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: 1. Create a DataFrame by reading in the 2019_Yellow_Taxi_Trip_Data.csv file. Examine the first 5 rows. 2. Find the dimensions (number of rows and number of columns) in the data. 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. 4. Isolate the fare_amount, tip_amount, tolls_amount, and total_amount for the longest trip by distance (trip_distance).

In [122…
```
import pandas as pd
#1. Create a DataFrame by reading in the 2019_Yellow_Taxi_Trip_Data.csv file. Exami
YellowTaxi = pd.read_csv('2019_Yellow_Taxi_Trip_Data.csv', nrows = 5)
YellowTaxi
```

Out[122…

	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

In [134…

```
#2. Find the dimensions (number of rows and number of columns) in the data.
YellowTaxi = pd.read_csv('2019_Yellow_Taxi_Trip_Data.csv')

YellowTaxi.shape
```

Out[134…

```
(10000, 18)
```

In [136…

```
#3. Using the data in the 2019_Yellow_Taxi_Trip_Data.csv file, calculate summary st

YellowTaxi[['fare_amount','tip_amount','tolls_amount','total_amount']].mean()
```

Out[136…

```
fare_amount     15.106313
tip_amount       2.634494
tolls_amount     0.623447
total_amount    22.564659
dtype: float64
```

In [137…

```
YellowTaxi[['fare_amount','tip_amount','tolls_amount','total_amount']].median()
```

Out[137…

```
fare_amount     10.0
tip_amount       2.0
tolls_amount     0.0
total_amount    16.3
dtype: float64
```

In [140…

```
YellowTaxi[['fare_amount','tip_amount','tolls_amount','total_amount']].quantile([0.
```

Out[140…

	fare_amount	tip_amount	tolls_amount	total_amount
0.01	3.000	0.000	0.00	6.3000
0.05	4.500	0.000	0.00	9.3000
0.50	10.000	2.000	0.00	16.3000
0.95	52.000	10.361	6.12	67.1075
0.99	62.005	15.860	6.12	82.4000

In [138…

```
YellowTaxi[['fare_amount','tip_amount','tolls_amount','total_amount']].max()
```

Out[138…

```
fare_amount     176.0
tip_amount       43.0
tolls_amount    612.0
total_amount    671.8
dtype: float64
```

In [133…

```
YellowTaxi[['fare_amount','tip_amount','tolls_amount','total_amount']].describe()
```

Out[133…

	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 [153…

```
YellowTaxi.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 18 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   vendorid               10000 non-null  int64
 1   tpep_pickup_datetime   10000 non-null  object
 2   tpep_dropoff_datetime  10000 non-null  object
 3   passenger_count        10000 non-null  int64
 4   trip_distance          10000 non-null  float64
 5   ratecodeid             10000 non-null  int64
 6   store_and_fwd_flag     10000 non-null  object
 7   pulocationid           10000 non-null  int64
 8   dolocationid           10000 non-null  int64
 9   payment_type           10000 non-null  int64
 10  fare_amount            10000 non-null  float64
 11  extra                  10000 non-null  float64
 12  mta_tax                10000 non-null  float64
 13  tip_amount             10000 non-null  float64
 14  tolls_amount           10000 non-null  float64
 15  improvement_surcharge  10000 non-null  float64
 16  total_amount           10000 non-null  float64
 17  congestion_surcharge   10000 non-null  float64
dtypes: float64(9), int64(6), object(3)
memory usage: 1.4+ MB
```

In [157…

```
#4. Isolate the fare_amount, tip_amount, tolls_amount, and total_amount for the lon

res = YellowTaxi.iloc[:,[10,13,14,16]]
res.describe()
```

Out[157…

	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 [159…

```
res.loc[YellowTaxi['trip_distance'].idxmax()]
```

Out[159…

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

Reflection

I think that I myself lack experimentation. The lesson is great and all but I wish that I could be better and I thinky by tackling a lot of examples and guidance it made it a lot easier to manipulate. More practice can ensure that I will improve

In [ ]: