In [1]:
```
import pandas as pd
```

In [46]:
```
meteorites = pd.read_csv('data/Meteorite_Landings.csv', nrows = 5)
meteorites
```

Out[46]:

	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 [7]:
```
# Series

meteorites.name
```

Out[7]:
```
0      Aachen
1      Aarhus
2        Abee
3    Acapulco
4     Achiras
Name: name, dtype: object
```

In [8]:
```
meteorites['name']
```

Out[8]:
```
0      Aachen
1      Aarhus
2        Abee
3    Acapulco
4     Achiras
Name: name, dtype: object
```

In [10]:
```
# Columns

meteorites.columns
```

Out[10]:
```
Index(['name', 'id', 'nametype', 'recclass', 'mass (g)', 'fall', 'year',
       'reclat', 'reclong', 'GeoLocation'],
      dtype='object')
```

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

Out[11]:
```
RangeIndex(start=0, stop=5, step=1)
```

In [42]:
```
# Using API

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 [41]:
```
payload
```

Out[41]:

```
[{'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'}}]
```


In [44]:
```
import pandas as pd

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

Out[44]:

	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 [48]:
```
meteorites = pd.read_csv('data/Meteorite_Landings.csv')
```

In [49]:
```
# How many rows and columns are there?

meteorites.shape
```

Out[49]:
```
(45716, 10)
```

In [50]:
```
# What are the column names?

meteorites.columns
```

Out[50]:
```
Index(['name', 'id', 'nametype', 'recclass', 'mass (g)', 'fall', 'year',
       'reclat', 'reclong', 'GeoLocation'],
      dtype='object')
```

In [52]:
```
# What type of data does each column currently hold?

meteorites.dtypes
```

Out[52]:
```
name           object
id              int64
nametype       object
recclass       object
mass (g)      float64
fall           object
year           object
reclat        float64
reclong       float64
GeoLocation    object
dtype: object
```

In [53]:
```
# What does the data look like?

meteorites.head(10) # First 10 values of meteorites.csv
```

Out[53]:

	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 [54]:
```
meteorites.tail(5) # Last 5 values of meteorites.csv
```

Out[54]:

	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 [55]:
```
# Get some inforamtion about the DataFrame
meteorites.info() # Keep in mind the missing values
```

```
<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 [68]:
```
meteorites.name
```

Out[68]:
```
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 [72]:
```
meteorites[["name", "fall", "mass (g)"]]
```

Out[72]:

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

45716 rows × 3 columns

In [73]:
```
# Selecting rows
meteorites[100:104]
```

Out[73]:

	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 [165…

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

Out[165…

	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 [162…

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

Out[162…

	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 [90]:

```
meteorites.iloc[: , -1] # value of last row and last column
```

Out[90]:
```
0             (50.775, 6.08333)
1          (56.18333, 10.23333)
2            (54.21667, -113.0)
3             (16.88333, -99.9)
4           (-33.16667, -64.95)
                   ...
45711         (29.037, 17.0185)
45712       (13.78333, 8.96667)
45713         (49.25, 17.66667)
45714       (49.78917, 41.5046)
45715    (33.98333, -115.68333)
Name: GeoLocation, Length: 45716, dtype: object
```

In [93]:
```
(meteorites['mass (g)'] > 50) & (meteorites.fall == 'Found')
```

Out[93]:
```
0        False
1         True
2         True
3         True
4         True
         ...
45711     True
45712     True
45713     True
45714     True
45715     True
Length: 45716, dtype: bool
```

In [94]:
```
meteorites[(meteorites['mass (g)'] > 50) & (meteorites.fall == 'Found')]
```

Out[94]:

	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 [96]:
```
meteorites.query("`mass (g)` > 1e6 and fall == 'Fell'")
```

Out[96]:

	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 [100…
```
# How many of the meteorites were found versus observed falling?

meteorites.fall.value_counts()
```

Out[100…
```
fall
Found    44609
Fell      1107
Name: count, dtype: int64
```

In [102…
```
meteorites.value_counts(subset = ['nametype', 'fall'], normalize = False) # return
```

Out[102…
```
nametype  fall
Valid     Found    44534
          Fell      1107
Relict    Found       75
Name: count, dtype: int64
```

In [107…
```
# What was the mass of the average meteorite?
type(float(meteorites['mass (g)'].mean()))
```

Out[107…
```
float
```

In [114…
```
meteorites['mass (g)'].quantile([0.01, 0.05, 0.5, 0.95, 0.99])
```

Out[114…
```
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 [112…
```
meteorites['mass (g)'].median()
```

Out[112… 32.6

In [110…
```
# What was the mass of the heaviest meteorite?
meteorites['mass (g)'].max()
```

Out[110… 60000000.0

In [150…
```
meteorites.loc[meteorites['mass (g)'].idxmax()]
```

Out[150…
```
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 [122…
```
# How many different types of meteorite classes are represented in this dataset?

meteorites.recclass.nunique()
```

Out[122… 466

In [124…
```
meteorites.recclass.unique()[:14]
```

Out[124…
```
array(['L5', 'H6', 'EH4', 'Acapulcoite', 'L6', 'LL3-6', 'H5', 'L',
       'Diogenite-pm', 'Unknown', 'H4', 'H', 'Iron, IVA', 'CR2-an'],
      dtype=object)
```

In [126…
```
meteorites.describe(include = 'all') # Summary of statistics
```

Out[126…

	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)

In [3]:
```
import pandas as pd
```

In [68]:
```
taxis = pd.read_csv('data/2019_Yellow_Taxi_Trip_Data.csv')
```

In [132…
```
# 1.) Create a DataFrame by reading in the 2019_Yellow_Taxi_Trip_Data.csv file. Exa

taxis = pd.DataFrame(df)
taxis.head(5)
```

Out[132…

	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 [40]:
```
# 2.) Find the dimensions (number of rows and number of columns) in the data.
taxis.shape
```

Out[40]: (10000, 18)

In [172…
```
# 3.) Using the data in the 2019_Yellow_Taxi_Trip_Data.csv file, calculate summary
new_taxis = taxis[['fare_amount', 'tip_amount', 'tolls_amount', 'total_amount']]

new_taxis.describe()
```

Out[172…

	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 [41]:
```
# 4.) Isolate the fare_amount, tip_amount, tolls_amount, and total_amount for the l
longest_trip = taxis.loc[taxis['trip_distance'].idxmax()]

fare_amount = longest_trip['fare_amount']
fare_amount
```

Out[41]: 176.0

In [179…
```
tip_amount = longest_trip['tip_amount']
tip_amount
```

Out[179… 18.29

In [178…
```
tolls_amount = longest_trip['tolls_amount']
tolls_amount
```

Out[178… 6.12

In [177…
```
total_amount = longest_trip['total_amount']
total_amount
```

Out[177… 201.21

In [ ]:
```
# Essay
# The progression of the lecture is good, it is challenging enough for us to get th
# It's just there's a lot of function needed in order do something great in the dat
# Overall, it's a nice practice especially we need to type it all.
```

In [ ]:
```
# Day 2
```

In [107…
```
mask = taxis.columns.str.contains('id$|store_and_fwd_flag', regex = True)
columns_to_drop = taxis.columns[mask]
columns_to_drop
```

Out[107…
```
Index(['vendorid', 'ratecodeid', 'store_and_fwd_flag', 'pulocationid',
       'dolocationid'],
      dtype='object')
```

In [21]:
```
taxis.drop(columns = columns_to_drop)
taxis.head()
```

Out[21]:

	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 [22]:
```
taxis = taxis.drop(columns = columns_to_drop)
taxis.head()
```

Out[22]:

	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 [70]:
```
# Renaming columns
taxis = taxis.rename(
     columns = {
         'tpep_pickup_datetime': 'pickup',
         'tpep_dropoff_datetime': 'dropoff'
     }
)
```

In [71]:
```
taxis
```

Out[71]:

	vendorid	pickup	dropoff	passenger_count	trip_distance	ratecodeid
0	2	2019-10-23T16:39:42.000	2019-10-23T17:14:10.000	1	7.93	1
1	1	2019-10-23T16:32:08.000	2019-10-23T16:45:26.000	1	2.00	1
2	2	2019-10-23T16:08:44.000	2019-10-23T16:21:11.000	1	1.36	1
3	2	2019-10-23T16:22:44.000	2019-10-23T16:43:26.000	1	1.00	1
4	2	2019-10-23T16:45:11.000	2019-10-23T16:58:49.000	1	1.96	1
...	...	...	...	...	...	...
9995	1	2019-10-23T17:39:59.000	2019-10-23T17:49:26.000	2	1.30	1
9996	1	2019-10-23T17:53:02.000	2019-10-23T18:00:45.000	1	1.40	1
9997	1	2019-10-23T17:07:16.000	2019-10-23T17:11:35.000	1	0.70	1
9998	1	2019-10-23T17:38:26.000	2019-10-23T17:49:28.000	2	2.50	1
9999	1	2019-10-23T17:22:14.000	2019-10-23T17:52:09.000	1	3.00	1

10000 rows × 18 columns

In [72]:
```
taxis[['pickup', 'dropoff']] = \
    taxis[['pickup', 'dropoff']].apply(pd.to_datetime)
taxis.dtypes
```

Out[72]:
```
vendorid                        int64
pickup                 datetime64[ns]
dropoff                datetime64[ns]
passenger_count                 int64
trip_distance                 float64
ratecodeid                      int64
store_and_fwd_flag             object
pulocationid                    int64
dolocationid                    int64
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
```

In [73]:
```
taxis = taxis.assign( # with assign(), it updates the original data frame
    elapsed_time = lambda x: x.dropoff - x.pickup, # 1 ------ the value was still s
    cost_before_tip = lambda x: x.total_amount - x.tip_amount,
    tip_pct = lambda x: x.tip_amount / x.cost_before_tip, # 2
    fees = lambda x: x.cost_before_tip - x.fare_amount, # 3
    avg_speed = lambda x: x.trip_distance.div(
        x.elapsed_time.dt.total_seconds() / 60 / 60
    ) # 4
)
```

In [63]:
```
taxis.sort_values('passenger_count', ascending = True).head()
```

Out[63]:

	vendorid	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distan
1663	1	2019-10-23T16:27:56.000	2019-10-23T17:25:20.000	0	6
8712	1	2019-10-23T16:12:14.000	2019-10-23T16:29:09.000	0	3
6015	1	2019-10-23T16:45:52.000	2019-10-23T17:38:38.000	0	8
9355	1	2019-10-24T08:19:11.000	2019-10-24T09:00:35.000	0	13
8225	1	2019-10-23T16:33:33.000	2019-10-23T16:48:37.000	0	1

In [74]:
```
taxis.sort_values(['passenger_count', 'pickup'], ascending = [True, False]).head()
```

Out[74]:

	vendorid	pickup	dropoff	passenger_count	trip_distance	ratecodeid	store_and_fv
9355	1	2019-10-24 08:19:11	2019-10-24 09:00:35	0	13.2	1	
9432	1	2019-10-23 17:54:48	2019-10-23 18:08:56	0	1.3	1	
9697	1	2019-10-23 17:52:10	2019-10-23 17:58:24	0	0.9	1	
9633	1	2019-10-23 17:52:02	2019-10-23 17:58:20	0	1.0	1	
9516	1	2019-10-23 17:48:43	2019-10-23 18:07:36	0	1.9	1	

5 rows × 23 columns

In [82]:
```
taxis.nlargest(3, 'passenger_count') # (number of largest in terms of , 'this param
```

Out[82]:

	vendorid	pickup	dropoff	passenger_count	trip_distance	ratecodeid	store_and_fwd
41	2	2019-10-23 16:12:20	2019-10-23 16:38:36	6	3.27	1	
42	2	2019-10-23 16:50:46	2019-10-23 16:57:37	6	0.80	1	
246	2	2019-10-23 16:41:32	2019-10-23 18:03:31	6	10.46	1	

3 rows × 23 columns

In [108…
```
# Exercise Part 2
```

In [109…
```
import pandas as pd
meteorites = pd.read_csv('data/Meteorite_Landings.csv')
```

In [110…
```
meteorites.head(5)
```

Out[110…

	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 [111…

```
# Rename column
meteorites = meteorites.rename(
    columns = {
        'mass (g)' : 'mass'
    }
)
meteorites # changed column name
```

Out[111…

	name	id	nametype	recclass	mass	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 [112…

```
# Drop all the latitude and longitude columns
meteorites = meteorites.drop(columns = ['reclat', 'reclong'])
meteorites
```

Out[112...

	name	id	nametype	recclass	mass	fall	year	GeoLocatio
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
...	...	...	...	...	...	...	...	.
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	01/01/1990 12:00:00 AM	(29.037 17.0185
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	01/01/1999 12:00:00 AM	(13.78333 8.96667
45713	Zlin	30410	Valid	H4	3.3	Found	01/01/1939 12:00:00 AM	(49.25 17.66667
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	01/01/2003 12:00:00 AM	(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

45716 rows × 8 columns

In [113...

```
# sort the result by mass in descending order

sorted_meteorites = meteorites.sort_values('mass', ascending = False)
sorted_meteorites
```

Out[113…

	name	id	nametype	recclass	mass	fall	year	GeoLocati
16392	Hoba	11890	Valid	Iron, IVB	60000000.0	Found	01/01/1920 12:00:00 AM	(-19.5833 17.9166
5373	Cape York	5262	Valid	Iron, IIIAB	58200000.0	Found	01/01/1818 12:00:00 AM	(76.1333 -64.9333
5365	Campo del Cielo	5247	Valid	Iron, IAB-MG	50000000.0	Found	12/22/1575 12:00:00 AM	(-27.4666 -60.5833
5370	Canyon Diablo	5257	Valid	Iron, IAB-MG	30000000.0	Found	01/01/1891 12:00:00 AM	(35.0 -111.0333
3455	Armanty	2335	Valid	Iron, IIIE	28000000.0	Found	01/01/1898 12:00:00 AM	(47.0, 88
...	...	...	...	...	...	...	...	
38282	Wei-hui-fu (a)	24231	Valid	Iron	NaN	Found	01/01/1931 12:00:00 AM	Na
38283	Wei-hui-fu (b)	24232	Valid	Iron	NaN	Found	01/01/1931 12:00:00 AM	Na
38285	Weiyuan	24233	Valid	Mesosiderite	NaN	Found	01/01/1978 12:00:00 AM	(35.2666 104.3166
41472	Yamato 792768	28117	Valid	CM2	NaN	Found	01/01/1979 12:00:00 AM	(-71 35.6666
45698	Zapata County	30393	Valid	Iron	NaN	Found	01/01/1930 12:00:00 AM	(27.0, -99

45716 rows × 8 columns

In [ ]: