In [521…

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

Out[521…

	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 [522…

```
meteorites['name']
```

Out[522…

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

In [523…

```
meteorites.columns
```

Out[523…

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

In [524…

```
meteorites.index
```

Out[524…

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

In [525…

```
import requests

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


if response.ok:
    payload = response.json()
```

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

In [526...

```
import pandas as pd

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

Out[526...

	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 [527...

```
import pandas as pd
meteorites = pd.read_csv('Meteorite_Landings.csv')
```

In [528...

```
meteorites.shape
```

Out[528...

```
(45716, 10)
```

In [529...

```
meteorites.columns
```

Out[529...

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

In [530...

```
meteorites.dtypes
```

Out[530...

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

In [531...

```
meteorites.head(10)
```

Out[531…

	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 [532…

```
meteorites.tail(5)
```

Out[532…

	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 [533…

```
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 [534…

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

Out[534…

	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 [535…

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

Out[535…

	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 [536…

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

Out[536…

	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 [537…

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

Out[537…

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

In [538…

```
meteorites.iloc[-1, [9]]
```

Out[538…

```
GeoLocation    (33.98333, -115.68333)
Name: 45715, dtype: object
```

In [539…

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

Out[539…

```
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 [540…

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

Out[540…

	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 [541…

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

Out[541…

	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 [542…

```
meteorites.fall.value_counts()
```

Out[542…

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

In [543…

```
meteorites.value_counts(subset=['nametype','fall'],normalize=False)
```

Out[543…

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

In [544…

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

Out[544…

```
np.float64(13278.078548601512)
```

In [545…

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

Out[545…

```
numpy.float64
```

In [546…

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

Out[546…

```
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 [547…

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

Out[547…

```
np.float64(32.6)
```

In [548…

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

Out[548…

```
np.float64(60000000.0)
```

In [549…

```python
meteorites.loc[meteorites['mass (g)'].idxmax()]
```

Out[549…

```
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 [550…

```python
meteorites.recclass.nunique()
```

Out[550…

```
466
```

In [551…

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

Out[551…

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

In [552…

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

Out[552…

```
array(['Aachen', 'Aarhus', 'Abee', 'Acapulco', 'Achiras', 'Adhi Kot',
       'Adzhi-Bogdo (stone)', 'Agen', 'Aguada', 'Aguila Blanca',
       'Aioun el Atrouss', 'Aïr', 'Aire-sur-la-Lys', 'Akaba'],
      dtype=object)
```

In [553…

```
meteorites.describe()
```

Out[553…

	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 [554…

```
meteorites.describe(include='all')
```

Out[554…

	name	id	nametype	recclass	mass (g)	fall	year	
count	45716	45716.000000	45716	45716	4.558500e+04	45716	45425	38401
unique	45716	NaN	2	466	NaN	2	266	
top	Zulu Queen	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

Exercise (Part 1)

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 columns4. . Isolate the fare_amount, tip_amount, tolls_amount, and total_amount for the longest trip by distance (trip_distance).

In [555…

```
#1
import pandas as pd
trip_data = pd.read_csv('2019_Yellow_Taxi_Trip_Data.csv', nrows=5)

trip_data
```

Out[555…

	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 [556…

```
#2
import pandas as pd
trip_data = pd.read_csv('2019_Yellow_Taxi_Trip_Data.csv')

trip_data.shape
```

Out[556…

```
(10000, 18)
```

In [557…

```
#3
trip_data.describe()
```

Out[557…

	vendorid	passenger_count	trip_distance	ratecodeid	pulocationid	dolocatic
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000
mean	1.633700	1.497700	3.015250	1.084200	166.900400	166.313
std	0.481817	1.139353	4.148063	0.418244	63.791288	68.525
min	1.000000	0.000000	0.000000	1.000000	1.000000	1.000
25%	1.000000	1.000000	0.920000	1.000000	132.000000	132.000
50%	2.000000	1.000000	1.500000	1.000000	162.000000	163.000
75%	2.000000	2.000000	2.760000	1.000000	234.000000	236.000
max	2.000000	6.000000	38.110000	5.000000	265.000000	265.000

In [558…

```
#3
trip_data.iloc[:,[10,13,14,16]].nunique()
```

Out[558…

```
fare_amount      199
tip_amount       657
tolls_amount      37
total_amount    1097
dtype: int64
```

In [559…

```
#3
trip_data.iloc[:,[10,13,14,16]].median()
```

Out[559…

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

In [560…

```
#3
trip_data.iloc[:,[10,13,14,16]].quantile([0.01,0.05,0.5,0.95,0.99])
```

Out[560…

	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 [561…

```
#3
trip_data.iloc[:,[10,13,14,16]].value_counts()
```

Out[561…

```
fare_amount  tip_amount  tolls_amount  total_amount
 7.5         0.00        0.0            11.80          126
 6.0         2.06        0.0            12.36          123
 6.5         0.00        0.0            10.80          119
 7.5         2.36        0.0            14.16          116
 10.5        0.00        0.0            14.80          115
                                                      ...
-5.5         0.00        0.0           -9.80             1
 12.5        3.95        0.0            19.75            1
-6.5         0.00        0.0           -8.30             1
-7.0         0.00        0.0           -11.30            1
-10.0        0.00        0.0           -11.80            1
Name: count, Length: 1878, dtype: int64
```

In [562…

```
trip_data.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 [563…

```
#4
trip_data.iloc[trip_data["trip_distance"].idxmax(), [10, 13, 14, 16]]
```

Out[563…

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

In [564…

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

Out[564…

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

In [565…

```
import pandas as pd
taxis = pd.read_csv('2019_Yellow_Taxi_Trip_Data.csv')
taxis.drop(columns=columns_to_drop)
```

Out[565…

	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	payme
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	
...	...	...	...	...	
9995	2019-10-23T17:39:59.000	2019-10-23T17:49:26.000	2	1.30	
9996	2019-10-23T17:53:02.000	2019-10-23T18:00:45.000	1	1.40	
9997	2019-10-23T17:07:16.000	2019-10-23T17:11:35.000	1	0.70	
9998	2019-10-23T17:38:26.000	2019-10-23T17:49:28.000	2	2.50	
9999	2019-10-23T17:22:14.000	2019-10-23T17:52:09.000	1	3.00	

10000 rows × 13 columns

In [566…

```
taxis = taxis.rename(
    columns={
        'tpep_pickup_datetime': 'pickup',
        'tpep_dropoff_datetime': 'dropoff'
    }
)
taxis.columns
```

Out[566…

```
Index(['vendorid', 'pickup', 'dropoff', 'passenger_count', 'trip_distance',
       'ratecodeid', 'store_and_fwd_flag', 'pulocationid', 'dolocationid',
       'payment_type', 'fare_amount', 'extra', 'mta_tax', 'tip_amount',
       'tolls_amount', 'improvement_surcharge', 'total_amount',
       'congestion_surcharge'],
      dtype='object')
```

In [567…

```
taxis.dtypes
```

Out[567…

```
vendorid                   int64
pickup                    object
dropoff                   object
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 [568…

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

Out[568…

```
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 [569…

```
taxis = taxis.assign(
    elapsed_time=lambda x: x.dropoff - x.pickup, #1
    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
)
taxis
```

Out[569…

	vendorid	pickup	dropoff	passenger_count	trip_distance	ratecodeid	store_and_fv
0	2	2019-10-23 16:39:42	2019-10-23 17:14:10	1	7.93	1	
1	1	2019-10-23 16:32:08	2019-10-23 16:45:26	1	2.00	1	
2	2	2019-10-23 16:08:44	2019-10-23 16:21:11	1	1.36	1	
3	2	2019-10-23 16:22:44	2019-10-23 16:43:26	1	1.00	1	
4	2	2019-10-23 16:45:11	2019-10-23 16:58:49	1	1.96	1	
...	...	...	...	...	...	...	
9995	1	2019-10-23 17:39:59	2019-10-23 17:49:26	2	1.30	1	
9996	1	2019-10-23 17:53:02	2019-10-23 18:00:45	1	1.40	1	
9997	1	2019-10-23 17:07:16	2019-10-23 17:11:35	1	0.70	1	
9998	1	2019-10-23 17:38:26	2019-10-23 17:49:28	2	2.50	1	
9999	1	2019-10-23 17:22:14	2019-10-23 17:52:09	1	3.00	1	

10000 rows × 23 columns

In [570…

```
taxis.dtypes
```

Out[570…

```
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
elapsed_time             timedelta64[ns]
cost_before_tip                  float64
tip_pct                          float64
fees                             float64
avg_speed                        float64
dtype: object
```

In [571…

```
taxis.sort_values(['trip_distance','ratecodeid'],ascending=[True, True]).head()
```

Out[571…

	vendorid	pickup	dropoff	passenger_count	trip_distance	ratecodeid	store_and_fv
517	2	2019-10-23 16:00:09	2019-10-23 16:00:20	3	0.0	1	
518	2	2019-10-23 16:00:09	2019-10-23 16:00:20	3	0.0	1	
996	1	2019-10-23 16:12:23	2019-10-23 16:13:36	1	0.0	1	
1066	1	2019-10-23 16:24:10	2019-10-23 16:24:49	1	0.0	1	
1485	2	2019-10-23 16:12:03	2019-10-23 16:12:07	1	0.0	1	

5 rows × 23 columns

In [572…

```
taxis.sort_values(['trip_distance','ratecodeid'],ascending=[False, True]).head()
```

Out[572…

	vendorid	pickup	dropoff	passenger_count	trip_distance	ratecodeid	store_and_fv
8338	2	2019-10-23 16:50:53	2019-10-24 15:32:55	1	38.11	5	
9965	2	2019-10-23 17:34:29	2019-10-23 18:48:00	1	37.86	2	
1656	2	2019-10-23 16:04:45	2019-10-23 19:11:40	3	37.57	2	
2237	2	2019-10-23 16:09:02	2019-10-23 17:40:37	1	28.41	1	
436	2	2019-10-23 16:43:22	2019-10-23 17:56:45	4	28.06	2	

5 rows × 23 columns

In [573…

```
#the first arguement ask for how many index to be shown and the next arguement is t
taxis.nlargest(5, 'elapsed_time')
```

Out[573…

	vendorid	pickup	dropoff	passenger_count	trip_distance	ratecodeid	store_and_fv
7576	2	2019-10-23 16:52:51	2019-10-24 16:51:44	1	3.75	1	
6902	2	2019-10-23 16:51:42	2019-10-24 16:50:22	1	11.19	1	
4975	2	2019-10-23 16:18:51	2019-10-24 16:17:30	1	0.70	1	
6550	2	2019-10-23 16:49:36	2019-10-24 16:47:40	1	2.54	1	
2585	2	2019-10-23 16:13:09	2019-10-24 16:11:08	1	3.67	1	

5 rows × 23 columns

Exercise (Part 2)

In [574...

```python
import pandas as pd
meteorite = pd.read_csv('Meteorite_Landings.csv')
meteorite.head()
```

Out[574...

	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 [575...

```python
meteorite = meteorite.rename(
    columns={
        'mass (g)': 'mass',
    }
)
meteorite.columns
```

Out[575...

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

In [576...

```python
mask = meteorite.columns.str.contains('reclat|reclong', regex=True)
columns_to_drop = meteorite.columns[mask]
columns_to_drop
```

Out[576...

```
Index(['reclat', 'reclong'], dtype='object')
```

In [577...

```python
meteorite = meteorite.drop(columns=columns_to_drop)
meteorite
```

Out[577…

	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 [578…

```
meteorite.sort_values(['mass'],ascending=[False]).head()
```

Out[578…

	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 [579…

```
taxis = taxis.set_index('pickup')
taxis.head(3)
```

Out[579…

	vendorid	dropoff	passenger_count	trip_distance	ratecodeid	store_and_fwd_flag
pickup						
2019-10-23 16:39:42	2	2019-10-23 17:14:10	1	7.93	1	N
2019-10-23 16:32:08	1	2019-10-23 16:45:26	1	2.00	1	N
2019-10-23 16:08:44	2	2019-10-23 16:21:11	1	1.36	1	N

3 rows × 22 columns

In [580…

```
taxis = taxis.sort_index()
taxis
```

Out[580…

pickup	vendorid	dropoff	passenger_count	trip_distance	ratecodeid	store_and_fwd_flag
2019-10-23 07:05:34	2	2019-10-23 08:03:16	3	14.68	1	N
2019-10-23 07:48:58	2	2019-10-23 07:52:09	1	0.67	1	N
2019-10-23 08:02:09	2	2019-10-24 07:42:32	1	8.38	1	N
2019-10-23 08:18:47	2	2019-10-23 08:36:05	1	2.39	1	N
2019-10-23 09:27:16	2	2019-10-23 09:33:13	2	1.11	1	N
...	...	...	...	...	...	...
2019-10-24 07:23:52	1	2019-10-24 08:08:52	1	0.00	1	Y
2019-10-24 07:29:52	2	2019-10-24 07:33:24	1	0.54	1	N
2019-10-24 07:58:31	1	2019-10-24 08:47:05	1	0.00	1	N
2019-10-24 08:07:45	2	2019-10-24 08:07:50	2	0.00	2	N
2019-10-24 08:19:11	1	2019-10-24 09:00:35	0	13.20	1	N

10000 rows × 22 columns

In [581…

```
taxis.sort_index(axis=1) #sorted by column names
```

Out[581…

pickup	avg_speed	congestion_surcharge	cost_before_tip	dolocationid	dropoff	elapsed_
2019-10-23 07:05:34	15.265165	0.0	51.8	181	2019-10-23 08:03:16	0 00:
2019-10-23 07:48:58	12.628272	2.5	8.8	141	2019-10-23 07:52:09	0 00:
2019-10-23 08:02:09	0.353989	2.5	36.3	33	2019-10-24 07:42:32	0 23:
2019-10-23 08:18:47	8.289017	2.5	16.8	237	2019-10-23 08:36:05	0 00:
2019-10-23 09:27:16	11.193277	0.0	7.8	41	2019-10-23 09:33:13	0 00:
...	...	...	...	...	...	
2019-10-24 07:23:52	0.000000	0.0	37.0	229	2019-10-24 08:08:52	0 00:
2019-10-24 07:29:52	9.169811	0.0	4.8	24	2019-10-24 07:33:24	0 00:
2019-10-24 07:58:31	0.000000	0.0	23.0	169	2019-10-24 08:47:05	0 00:
2019-10-24 08:07:45	0.000000	2.5	55.3	163	2019-10-24 08:07:50	0 00:
2019-10-24 08:19:11	19.130435	0.0	42.8	223	2019-10-24 09:00:35	0 00:

10000 rows × 22 columns

In [582…

```
taxis.sort_index(axis=0) #sorted by index
```

Out[582…

	vendorid	dropoff	passenger_count	trip_distance	ratecodeid	store_and_fwd_flag
pickup						
2019-10-23 07:05:34	2	2019-10-23 08:03:16	3	14.68	1	N
2019-10-23 07:48:58	2	2019-10-23 07:52:09	1	0.67	1	N
2019-10-23 08:02:09	2	2019-10-24 07:42:32	1	8.38	1	N
2019-10-23 08:18:47	2	2019-10-23 08:36:05	1	2.39	1	N
2019-10-23 09:27:16	2	2019-10-23 09:33:13	2	1.11	1	N
...	...	...	...	...	...	..
2019-10-24 07:23:52	1	2019-10-24 08:08:52	1	0.00	1	Y
2019-10-24 07:29:52	2	2019-10-24 07:33:24	1	0.54	1	N
2019-10-24 07:58:31	1	2019-10-24 08:47:05	1	0.00	1	N
2019-10-24 08:07:45	2	2019-10-24 08:07:50	2	0.00	2	N
2019-10-24 08:19:11	1	2019-10-24 09:00:35	0	13.20	1	N

10000 rows × 22 columns

In [583…

```
taxis.loc['2019-10']
```

Out[583…

	vendorid	dropoff	passenger_count	trip_distance	ratecodeid	store_and_fwd_flag
pickup						
2019-10-23 07:05:34	2	2019-10-23 08:03:16	3	14.68	1	N
2019-10-23 07:48:58	2	2019-10-23 07:52:09	1	0.67	1	N
2019-10-23 08:02:09	2	2019-10-24 07:42:32	1	8.38	1	N
2019-10-23 08:18:47	2	2019-10-23 08:36:05	1	2.39	1	N
2019-10-23 09:27:16	2	2019-10-23 09:33:13	2	1.11	1	N
...	...	...	...	...	...	..
2019-10-24 07:23:52	1	2019-10-24 08:08:52	1	0.00	1	Y
2019-10-24 07:29:52	2	2019-10-24 07:33:24	1	0.54	1	N
2019-10-24 07:58:31	1	2019-10-24 08:47:05	1	0.00	1	N
2019-10-24 08:07:45	2	2019-10-24 08:07:50	2	0.00	2	N
2019-10-24 08:19:11	1	2019-10-24 09:00:35	0	13.20	1	N

10000 rows × 22 columns

In [584…

```
taxis['2019-10-23 07:45':'2019-10-23 08']
```

Out[584…

	vendorid	dropoff	passenger_count	trip_distance	ratecodeid	store_and_fwd_flag
pickup						
2019-10-23 07:48:58	2	2019-10-23 07:52:09	1	0.67	1	N
2019-10-23 08:02:09	2	2019-10-24 07:42:32	1	8.38	1	N
2019-10-23 08:18:47	2	2019-10-23 08:36:05	1	2.39	1	N

3 rows × 22 columns

In [585…

```python
taxis = taxis.reset_index()
taxis.head(5)
```

Out[585…

	pickup	vendorid	dropoff	passenger_count	trip_distance	ratecodeid	store_and_fwd_f
0	2019-10-23 07:05:34	2	2019-10-23 08:03:16	3	14.68	1	
1	2019-10-23 07:48:58	2	2019-10-23 07:52:09	1	0.67	1	
2	2019-10-23 08:02:09	2	2019-10-24 07:42:32	1	8.38	1	
3	2019-10-23 08:18:47	2	2019-10-23 08:36:05	1	2.39	1	
4	2019-10-23 09:27:16	2	2019-10-23 09:33:13	2	1.11	1	

5 rows × 23 columns

Exercise (Part 3)

In [606…

```python
import pandas as pd
meteorite = pd.read_csv('Meteorite_Landings.csv')
meteorite.head()
```

Out[606…

	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 [607…

```
meteorite['year'] = meteorite['year'].str.slice(6,11)
meteorite
```

Out[607…

	name	id	nametype	recclass	mass (g)	fall	year	reclat	recl
0	Aachen	1	Valid	L5	21.0	Fell	1880	50.77500	6.08
1	Aarhus	2	Valid	H6	720.0	Fell	1951	56.18333	10.23
2	Abee	6	Valid	EH4	107000.0	Fell	1952	54.21667	-113.00
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	1976	16.88333	-99.90
4	Achiras	370	Valid	L6	780.0	Fell	1902	-33.16667	-64.95
...	...	...	...	...	...	...	...	...	
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	1990	29.03700	17.01
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	1999	13.78333	8.96
45713	Zlin	30410	Valid	H4	3.3	Found	1939	49.25000	17.66
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	2003	49.78917	41.50
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	1976	33.98333	-115.68

45716 rows × 10 columns

In [608…

```
meteorite.dtypes
```

Out[608…

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

In [609…

```
meteorite['year'] = meteorite['year'].apply(pd.to_numeric)
meteorite.dtypes
```

Out[609…

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

In [610…

```
meteorite['Fell before 1970']= (meteorite.year < 1970) & (meteorite.fall == 'Fell')
meteorite
```

Out[610…

	name	id	nametype	recclass	mass (g)	fall	year	reclat	re
0	Aachen	1	Valid	L5	21.0	Fell	1880.0	50.77500	6.
1	Aarhus	2	Valid	H6	720.0	Fell	1951.0	56.18333	10.
2	Abee	6	Valid	EH4	107000.0	Fell	1952.0	54.21667	-113.
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	1976.0	16.88333	-99.
4	Achiras	370	Valid	L6	780.0	Fell	1902.0	-33.16667	-64.
...	...	...	...	...	...	...	...	...	
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	1990.0	29.03700	17.
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	1999.0	13.78333	8.
45713	Zlin	30410	Valid	H4	3.3	Found	1939.0	49.25000	17.
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	2003.0	49.78917	41.
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	1976.0	33.98333	-115.

45716 rows × 11 columns

In [611…

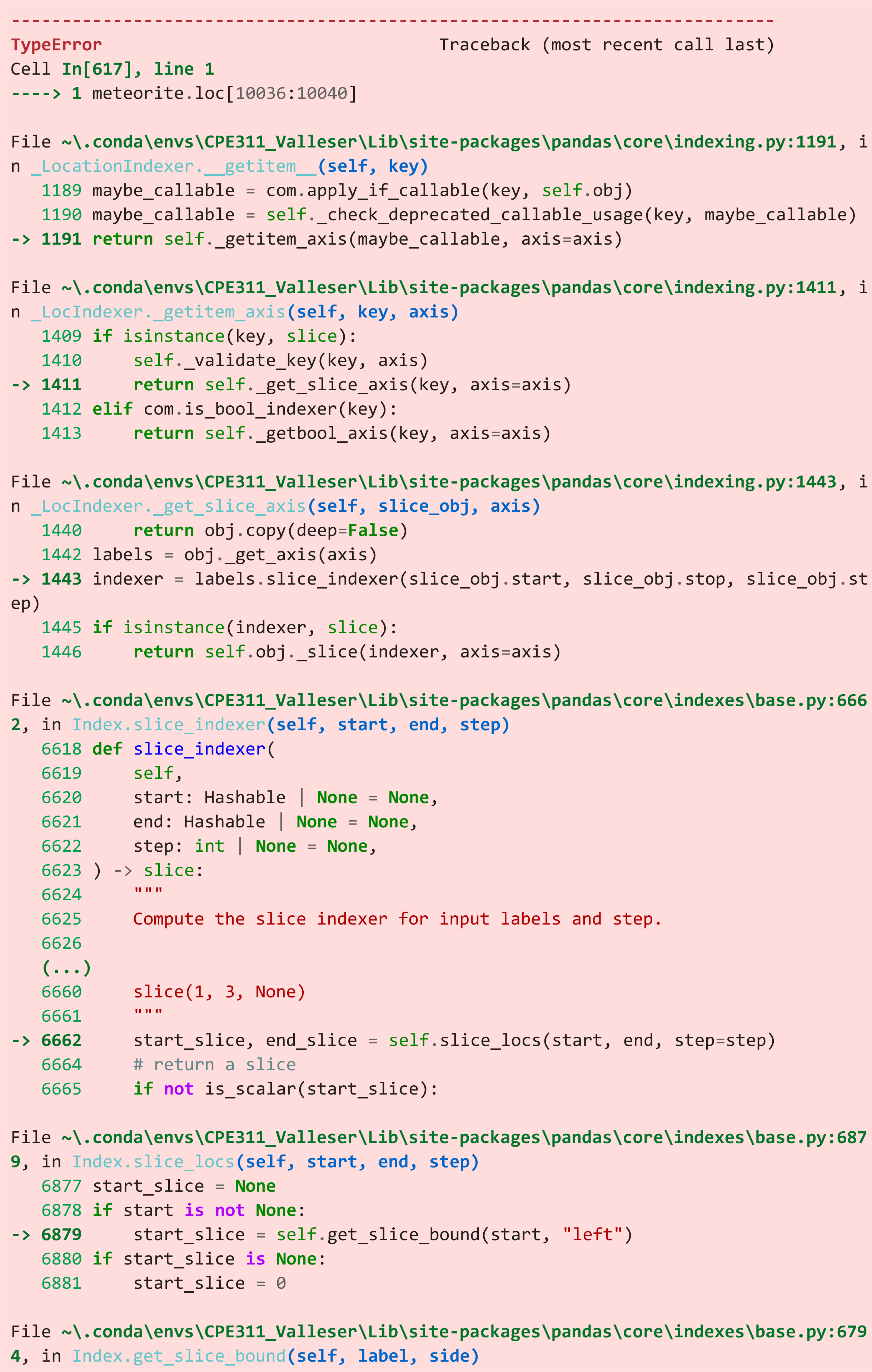
```
meteorite = meteorite.set_index('id')
```

In [612…

```
meteorite = meteorite.sort_index()
```

In [617…

```
meteorite.loc[10036:10040]
```



```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[617], line 1
----> 1 meteorite.loc[10036:10040]

File ~\.conda\envs\CPE311_Valleser\Lib\site-packages\pandas\core\indexing.py:1191, i
n _LocationIndexer.__getitem__(self, key)
   1189 maybe_callable = com.apply_if_callable(key, self.obj)
   1190 maybe_callable = self._check_deprecated_callable_usage(key, maybe_callable)
-> 1191 return self._getitem_axis(maybe_callable, axis=axis)

File ~\.conda\envs\CPE311_Valleser\Lib\site-packages\pandas\core\indexing.py:1411, i
n _LocIndexer._getitem_axis(self, key, axis)
   1409 if isinstance(key, slice):
   1410     self._validate_key(key, axis)
-> 1411     return self._get_slice_axis(key, axis=axis)
   1412 elif com.is_bool_indexer(key):
   1413     return self._getbool_axis(key, axis=axis)

File ~\.conda\envs\CPE311_Valleser\Lib\site-packages\pandas\core\indexing.py:1443, i
n _LocIndexer._get_slice_axis(self, slice_obj, axis)
   1440     return obj.copy(deep=False)
   1442 labels = obj._get_axis(axis)
-> 1443 indexer = labels.slice_indexer(slice_obj.start, slice_obj.stop, slice_obj.st
ep)
   1445 if isinstance(indexer, slice):
   1446     return self.obj._slice(indexer, axis=axis)

File ~\.conda\envs\CPE311_Valleser\Lib\site-packages\pandas\core\indexes\base.py:666
2, in Index.slice_indexer(self, start, end, step)
   6618 def slice_indexer(
   6619     self,
   6620     start: Hashable | None = None,
   6621     end: Hashable | None = None,
   6622     step: int | None = None,
   6623 ) -> slice:
   6624     """
   6625     Compute the slice indexer for input labels and step.
   6626 
   (...)
   6660     slice(1, 3, None)
   6661     """
-> 6662     start_slice, end_slice = self.slice_locs(start, end, step=step)
   6664     # return a slice
   6665     if not is_scalar(start_slice):

File ~\.conda\envs\CPE311_Valleser\Lib\site-packages\pandas\core\indexes\base.py:687
9, in Index.slice_locs(self, start, end, step)
   6877 start_slice = None
   6878 if start is not None:
-> 6879     start_slice = self.get_slice_bound(start, "left")
   6880 if start_slice is None:
   6881     start_slice = 0

File ~\.conda\envs\CPE311_Valleser\Lib\site-packages\pandas\core\indexes\base.py:679
4, in Index.get_slice_bound(self, label, side)
```

```
   6790 original_label = label
   6792 # For datetime indices label may be a string that has to be converted
   6793 # to datetime boundary according to its resolution.
-> 6794 label = self._maybe_cast_slice_bound(label, side)
   6796 # we need to look up the label
   6797 try:

File ~\.conda\envs\CPE311_Valleser\Lib\site-packages\pandas\core\indexes\base.py:6727, in Index._maybe_cast_slice_bound(self, label, side)
   6725 # reject them, if index does not contain label
   6726 if (is_float(label) or is_integer(label)) and label not in self:
-> 6727     self._raise_invalid_indexer("slice", label)
   6729 return label

File ~\.conda\envs\CPE311_Valleser\Lib\site-packages\pandas\core\indexes\base.py:4301, in Index._raise_invalid_indexer(self, form, key, reraise)
   4299 if reraise is not lib.no_default:
   4300     raise TypeError(msg) from reraise
-> 4301 raise TypeError(msg)

TypeError: cannot do slice indexing on Index with these indexers [10036] of type int
```

In [ ]:

In [ ]: