

In [268... `import pandas as pd`

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

In [269... `mask = taxis.columns.str.contains('id$|store_and_fwd_flag', regex = True)`  
`columns_to_drop = taxis.columns[mask]`  
`columns_to_drop`

Out[269... `Index(['vendorid', 'ratecodeid', 'store_and_fwd_flag', 'pulocationid',  
 'dolocationid'],  
 dtype='object')`

In [270... `taxis = taxis.drop(columns = columns_to_drop)`  
`taxis.head()`

Out[270... 

	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	payment_type	fare_amount	extra	mta_tax	tip_amount
0	2019-10-23T16:39:42.000	2019-10-23T17:14:10.000	1	7.93	1	29.5	1.0	0.5	7.98
1	2019-10-23T16:32:08.000	2019-10-23T16:45:26.000	1	2.00	1	10.5	1.0	0.5	0.00
2	2019-10-23T16:08:44.000	2019-10-23T16:21:11.000	1	1.36	1	9.5	1.0	0.5	2.00
3	2019-10-23T16:22:44.000	2019-10-23T16:43:26.000	1	1.00	1	13.0	1.0	0.5	4.32
4	2019-10-23T16:45:11.000	2019-10-23T16:58:49.000	1	1.96	1	10.5	1.0	0.5	0.50

## Renaming Columns

In [271... `taxis = taxis.rename(  
 columns = {  
 'tpep_pickup_datetime': 'pickup',  
 'tpep_dropoff_datetime': 'dropoff'  
 }  
)`

In [272... `taxis.dtypes`

Out[272... 

pickup	object
dropoff	object
passenger_count	int64
trip_distance	float64
payment_type	int64
fare_amount	float64
extra	float64
mta_tax	float64
tip_amount	float64
tolls_amount	float64
improvement_surcharge	float64
total_amount	float64
congestion_surcharge	float64
dtype:	object

In [273... `taxis[['pickup', 'dropoff']] = taxis[['pickup', 'dropoff']].apply(pd.to_datetime)`  
`taxis.dtypes`

```
Out[273...] pickup          datetime64[ns]
dropoff        datetime64[ns]
passenger_count int64
trip_distance   float64
payment_type    int64
fare_amount     float64
extra           float64
mta_tax         float64
tip_amount      float64
tolls_amount    float64
improvement_surcharge float64
total_amount    float64
congestion_surcharge float64
dtype: object
```

## Create New Columns

```
In [274...] taxi = taxi.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
)
```

taxi['elapsed\_time'] = taxi['dropoff']- taxi['pickup'] taxi

```
In [275...] taxi.dtypes
```

```
Out[275...] pickup          datetime64[ns]
dropoff        datetime64[ns]
passenger_count int64
trip_distance   float64
payment_type    int64
fare_amount     float64
extra           float64
mta_tax         float64
tip_amount      float64
tolls_amount    float64
improvement_surcharge float64
total_amount    float64
congestion_surcharge float64
elapsed_time    timedelta64[ns]
cost_before_tip float64
tip_pct         float64
fees            float64
avg_speed       float64
dtype: object
```

```
In [276...] taxi.sort_values(['trip_distance', 'fees'], ascending = [False, True]).head()
```

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement_surcharge
8338	2019-10-23 16:50:53	2019-10-24 15:32:55	1	38.11	1	176.0	0.0	0.5	18.29	6.12	
9965	2019-10-23 17:34:29	2019-10-23 18:48:00	1	37.86	2	52.0	4.5	0.5	0.00	6.12	
1656	2019-10-23 16:04:45	2019-10-23 19:11:40	3	37.57	1	52.0	4.5	0.5	13.18	6.12	
2237	2019-10-23 16:09:02	2019-10-23 17:40:37	1	28.41	1	87.5	1.0	0.5	0.00	6.12	
436	2019-10-23 16:43:22	2019-10-23 17:56:45	4	28.06	1	52.0	4.5	0.5	13.18	6.12	

```
In [277...] taxi.head()
```

Out [277...

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement
<b>0</b>	2019-10-23 16:39:42	2019-10-23 17:14:10	1	7.93	1	29.5	1.0	0.5	7.98	6.12	
<b>1</b>	2019-10-23 16:32:08	2019-10-23 16:45:26	1	2.00	1	10.5	1.0	0.5	0.00	0.00	
<b>2</b>	2019-10-23 16:08:44	2019-10-23 16:21:11	1	1.36	1	9.5	1.0	0.5	2.00	0.00	
<b>3</b>	2019-10-23 16:22:44	2019-10-23 16:43:26	1	1.00	1	13.0	1.0	0.5	4.32	0.00	
<b>4</b>	2019-10-23 16:45:11	2019-10-23 16:58:49	1	1.96	1	10.5	1.0	0.5	0.50	0.00	

In [278... `taxis.nlargest(3, 'elapsed_time')`

Out [278...

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement
<b>7576</b>	2019-10-23 16:52:51	2019-10-24 16:51:44	1	3.75	1	17.5	1.0	0.5	0.0	0.0	
<b>6902</b>	2019-10-23 16:51:42	2019-10-24 16:50:22	1	11.19	2	39.5	1.0	0.5	0.0	0.0	
<b>4975</b>	2019-10-23 16:18:51	2019-10-24 16:17:30	1	0.70	2	7.0	1.0	0.5	0.0	0.0	

In [279... `taxis.nlargest(5, 'fees') # for finding the largest rows, '.nsmallest()`

Out [279...

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement
<b>449</b>	2019-10-23 16:43:02	2019-10-23 18:02:40	1	17.00	3	52.0	7.0	0.5	0.00	612.00	
<b>8898</b>	2019-10-23 16:59:05	2019-10-23 18:24:26	2	18.90	2	52.0	7.0	0.5	0.00	21.12	
<b>3354</b>	2019-10-23 16:23:19	2019-10-23 17:10:00	1	10.01	1	95.0	0.0	0.5	24.66	25.00	
<b>9758</b>	2019-10-23 17:20:50	2019-10-23 18:58:16	1	19.50	1	96.0	1.0	0.0	37.25	27.00	
<b>3486</b>	2019-10-23 16:40:51	2019-10-23 17:58:55	1	16.96	2	79.5	1.0	0.0	0.00	21.00	

## Exercise 2

In [238... `# Loading the data`  
`meteorite = pd.read_csv('Meteorite_Landings.csv')`

In [239... `# Renaming the mass (g) column`  
`meteorite = meteorite.rename(`  
 `columns = {`  
 `'mass (g)': 'mass'`  
 `}`  
`)`  
`meteorite`

Out[239...

	name	id	nametype	recclass	mass	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)
...	...	...	...	...	...	...	...	...	...	...
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)

45716 rows × 10 columns

In [240...

```
# Drop Latitude and Longitude
mask = meteorite.columns.str.contains('GeoLocation', regex = True)
columns_to_drop = meteorite.columns[mask]
columns_to_drop
```

Out[240...

Index(['GeoLocation'], dtype='object')

In [241...

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

Out[241...

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

In [242...

```
# Sort the result by mass in descending
sort = meteorite.sort_values(['mass'], ascending = [False])
sort
```

Out[242...

	name	id	nametype	recclass	mass	fall	year	reclat	reclong
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	35.05000	-111.03333
3455	Armanty	2335	Valid	Iron, IIIE	28000000.0	Found	01/01/1898 12:00:00 AM	47.00000	88.00000
...	...	...	...	...	...	...	...	...	...
38282	Wei-hui-fu (a)	24231	Valid	Iron	NaN	Found	01/01/1931 12:00:00 AM	NaN	NaN
38283	Wei-hui-fu (b)	24232	Valid	Iron	NaN	Found	01/01/1931 12:00:00 AM	NaN	NaN
38285	Weiyuan	24233	Valid	Mesosiderite	NaN	Found	01/01/1978 12:00:00 AM	35.26667	104.31667
41472	Yamato 792768	28117	Valid	CM2	NaN	Found	01/01/1979 12:00:00 AM	-71.50000	35.66667
45698	Zapata County	30393	Valid	Iron	NaN	Found	01/01/1930 12:00:00 AM	27.00000	-99.00000

45716 rows × 9 columns

# Working with Index

## Setting and Soritng the Index

In [280...

```
taxis = taxis.set_index('pickup') # ISO Formatting of date yyyy-mm-dd
taxis.head(3)
```

Out[280...

	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement_su
pickup										
2019-10-23 16:39:42	2019-10-23 17:14:10	1	7.93	1	29.5	1.0	0.5	7.98	6.12	
2019-10-23 16:32:08	2019-10-23 16:45:26	1	2.00	1	10.5	1.0	0.5	0.00	0.00	
2019-10-23 16:08:44	2019-10-23 16:21:11	1	1.36	1	9.5	1.0	0.5	2.00	0.00	

In [281...

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

Out[281...

		avg_speed	congestion_surcharge	cost_before_tip	dropoff	elapsed_time	extra	fare_amount	fees	improvement_surcharge	mt
pickup											
2019-10-23 16:39:42		13.804642	2.5	39.92	2019-10-23 17:14:10	0 days 00:34:28	1.0	29.5	10.42		0.3
2019-10-23 16:32:08		9.022556	0.0	12.30	2019-10-23 16:45:26	0 days 00:13:18	1.0	10.5	1.80		0.3
2019-10-23 16:08:44		6.554217	2.5	13.80	2019-10-23 16:21:11	0 days 00:12:27	1.0	9.5	4.30		0.3
2019-10-23 16:22:44		2.898551	2.5	17.30	2019-10-23 16:43:26	0 days 00:20:42	1.0	13.0	4.30		0.3
2019-10-23 16:45:11		8.625917	2.5	14.80	2019-10-23 16:58:49	0 days 00:13:38	1.0	10.5	4.30		0.3
...	...	...	...	...	...	...	...	...	...	...	...
2019-10-23 17:39:59		8.253968	2.5	12.30	2019-10-23 17:49:26	0 days 00:09:27	3.5	8.0	4.30		0.3
2019-10-23 17:53:02		10.885529	2.5	12.30	2019-10-23 18:00:45	0 days 00:07:43	3.5	8.0	4.30		0.3
2019-10-23 17:07:16		9.729730	0.0	6.80	2019-10-23 17:11:35	0 days 00:04:19	1.0	5.0	1.80		0.3
2019-10-23 17:38:26		13.595166	0.0	11.80	2019-10-23 17:49:28	0 days 00:11:02	1.0	10.0	1.80		0.3
2019-10-23 17:22:14		6.016713	2.5	23.30	2019-10-23 17:52:09	0 days 00:29:55	3.5	19.0	4.30		0.3

10000 rows × 17 columns



In [282...

```
taxi.sort_index(axis = 0) # sort by column names thru index
```

Out [282...

	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement_su
pickup										
2019-10-23 07:05:34	2019-10-23 08:03:16	3	14.68	1	50.0	1.0	0.5	4.0	0.0	
2019-10-23 07:48:58	2019-10-23 07:52:09	1	0.67	2	4.5	1.0	0.5	0.0	0.0	
2019-10-23 08:02:09	2019-10-24 07:42:32	1	8.38	1	32.0	1.0	0.5	5.5	0.0	
2019-10-23 08:18:47	2019-10-23 08:36:05	1	2.39	2	12.5	1.0	0.5	0.0	0.0	
2019-10-23 09:27:16	2019-10-23 09:33:13	2	1.11	2	6.0	1.0	0.5	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...
2019-10-24 07:23:52	2019-10-24 08:08:52	1	0.00	1	36.2	0.0	0.5	0.0	0.0	
2019-10-24 07:29:52	2019-10-24 07:33:24	1	0.54	2	4.0	0.0	0.5	0.0	0.0	
2019-10-24 07:58:31	2019-10-24 08:47:05	1	0.00	1	22.2	0.0	0.5	0.0	0.0	
2019-10-24 08:07:45	2019-10-24 08:07:50	2	0.00	2	52.0	0.0	0.5	0.0	0.0	
2019-10-24 08:19:11	2019-10-24 09:00:35	0	13.20	2	42.0	0.0	0.5	0.0	0.0	

10000 rows × 17 columns



In [283...

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

```

-----
KeyError                                Traceback (most recent call last)
Cell In[283], line 1
----> 1 taxi['2019-10-23 07:45': '2019-10-23 08']

File ~\.conda\envs\CPE311_Corpuz\Lib\site-packages\pandas\core\frame.py:4085, in DataFrame.__getitem__(self, key)
    4083 # Do we have a slicer (on rows)?
    4084 if isinstance(key, slice):
-> 4085     return self._getitem_slice(key)
    4087 # Do we have a (boolean) DataFrame?
    4088 if isinstance(key, DataFrame):

File ~\.conda\envs\CPE311_Corpuz\Lib\site-packages\pandas\core\generic.py:4349, in NDFrame._getitem_slice(self, key)
    4344 """
    4345 _getitem_ for the case where the key is a slice object.
    4346 """
    4347 # _convert_slice_indexer to determine if this slice is positional
    4348 # or label based, and if the latter, convert to positional
-> 4349 slobj = self.index._convert_slice_indexer(key, kind="getitem")
    4350 if isinstance(slobj, np.ndarray):
    4351     # reachable with DatetimeIndex
    4352     indexer = lib.maybe_indices_to_slice(
    4353         slobj.astype(np.intp, copy=False), len(self)
    4354     )

File ~\.conda\envs\CPE311_Corpuz\Lib\site-packages\pandas\core\indexes\base.py:4281, in Index._convert_slice_indexer(self, key, kind)
    4279 indexer = key
    4280 else:
-> 4281     indexer = self.slice_indexer(start, stop, step)
    4283 return indexer

File ~\.conda\envs\CPE311_Corpuz\Lib\site-packages\pandas\core\indexes\datetime.py:697, in DatetimeIndex.slice_indexer(self, start, end, step)
    694 in_index &= (end_casted == self).any()
    696 if not in_index:
--> 697     raise KeyError(
    698         "Value based partial slicing on non-monotonic DatetimeIndexes "
    699         "with non-existing keys is not allowed.",
    700     )
    701 indexer = mask.nonzero()[0][::step]
    702 if len(indexer) == len(self):

KeyError: 'Value based partial slicing on non-monotonic DatetimeIndexes with non-existing keys is not allowed.'

```

In [284... taxi.loc['2019-10-23 08']

Out[284...      dropoff   passenger\_count   trip\_distance   payment\_type   fare\_amount   extra   mta\_tax   tip\_amount   tolls\_amount   improvement\_su

pickup										
2019-10-23 08:02:09	2019-10-24 07:42:32	1	8.38	1	32.0	1.0	0.5	5.5	0.0	
2019-10-23 08:18:47	2019-10-23 08:36:05	1	2.39	2	12.5	1.0	0.5	0.0	0.0	

In [105... taxi = taxi.reset\_index()  
taxi.head()



Out[105...

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement
0	2019-10-23 16:39:42	2019-10-23 17:14:10	1	7.93	1	29.5	1.0	0.5	7.98	6.12	
1	2019-10-23 16:32:08	2019-10-23 16:45:26	1	2.00	1	10.5	1.0	0.5	0.00	0.00	
2	2019-10-23 16:08:44	2019-10-23 16:21:11	1	1.36	1	9.5	1.0	0.5	2.00	0.00	
3	2019-10-23 16:22:44	2019-10-23 16:43:26	1	1.00	1	13.0	1.0	0.5	4.32	0.00	
4	2019-10-23 16:45:11	2019-10-23 16:58:49	1	1.96	1	10.5	1.0	0.5	0.50	0.00	

Exercise 3

In [243...

```
# Update the year column to only contain the year
meteorite['year'] = meteorite.year.str.slice(6, 11)
meteorite
```

Out[243...

	name	id	nametype	recclass	mass	fall	year	reclat	reclong
0	Aachen	1	Valid	L5	21.0	Fell	1880	50.77500	6.08333
1	Aarhus	2	Valid	H6	720.0	Fell	1951	56.18333	10.23333
2	Abee	6	Valid	EH4	107000.0	Fell	1952	54.21667	-113.00000
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	1976	16.88333	-99.90000
4	Achiras	370	Valid	L6	780.0	Fell	1902	-33.16667	-64.95000
...	...	...	...	...	...	...	...	...	...
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	1990	29.03700	17.01850
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	1999	13.78333	8.96667
45713	Zlin	30410	Valid	H4	3.3	Found	1939	49.25000	17.66667
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	2003	49.78917	41.50460
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	1976	33.98333	-115.68333

45716 rows × 9 columns

In [247...

```
# Convert it to numeric data type
meteorite['year'] = pd.to_numeric(meteorite['year'], errors='coerce')
meteorite
```

Out [247...

	name	id	nametype	recclass	mass	fall	year	reclat	reclong
0	Aachen	1	Valid	L5	21.0	Fell	1880.0	50.77500	6.08333
1	Aarhus	2	Valid	H6	720.0	Fell	1951.0	56.18333	10.23333
2	Abee	6	Valid	EH4	107000.0	Fell	1952.0	54.21667	-113.00000
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	1976.0	16.88333	-99.90000
4	Achiras	370	Valid	L6	780.0	Fell	1902.0	-33.16667	-64.95000
...	...	...	...	...	...	...	...	...	...
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	1990.0	29.03700	17.01850
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	1999.0	13.78333	8.96667
45713	Zlin	30410	Valid	H4	3.3	Found	1939.0	49.25000	17.66667
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	2003.0	49.78917	41.50460
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	1976.0	33.98333	-115.68333

45716 rows × 9 columns

In [253...

```
# Create a new column indicating whether the meteorite was observed falling before 1970
meteorite['Before 1970'] = meteorite['year'] < 1970
meteorite
```

Out [253...

	name	id	nametype	recclass	mass	fall	year	reclat	reclong	Before 1970
0	Aachen	1	Valid	L5	21.0	Fell	1880.0	50.77500	6.08333	True
1	Aarhus	2	Valid	H6	720.0	Fell	1951.0	56.18333	10.23333	True
2	Abee	6	Valid	EH4	107000.0	Fell	1952.0	54.21667	-113.00000	True
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	1976.0	16.88333	-99.90000	False
4	Achiras	370	Valid	L6	780.0	Fell	1902.0	-33.16667	-64.95000	True
...	...	...	...	...	...	...	...	...	...	...
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	1990.0	29.03700	17.01850	False
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	1999.0	13.78333	8.96667	False
45713	Zlin	30410	Valid	H4	3.3	Found	1939.0	49.25000	17.66667	True
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	2003.0	49.78917	41.50460	False
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	1976.0	33.98333	-115.68333	False

45716 rows × 10 columns

In [255...

```
# Setting the index to the id column
meteorite = meteorite.set_index('id')
meteorite
```

Out[255...

	name	nametype	recclass	mass	fall	year	reclat	reclong	Before 1970
id									
1	Aachen	Valid	L5	21.0	Fell	1880.0	50.77500	6.08333	True
2	Aarhus	Valid	H6	720.0	Fell	1951.0	56.18333	10.23333	True
6	Abee	Valid	EH4	107000.0	Fell	1952.0	54.21667	-113.00000	True
10	Acapulco	Valid	Acapulcoite	1914.0	Fell	1976.0	16.88333	-99.90000	False
370	Achiras	Valid	L6	780.0	Fell	1902.0	-33.16667	-64.95000	True
...	...	...	...	...	...	...	...	...	...
31356	Zillah 002	Valid	Eucrite	172.0	Found	1990.0	29.03700	17.01850	False
30409	Zinder	Valid	Pallasite, ungrouped	46.0	Found	1999.0	13.78333	8.96667	False
30410	Zlin	Valid	H4	3.3	Found	1939.0	49.25000	17.66667	True
31357	Zubkovsky	Valid	L6	2167.0	Found	2003.0	49.78917	41.50460	False
30414	Zulu Queen	Valid	L3.7	200.0	Found	1976.0	33.98333	-115.68333	False

45716 rows × 9 columns

In [256...

```
# Extract all the rows with IDs
meteorite = meteorite.sort_index()
```

In [257...

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

Out[257...

	name	nametype	recclass	mass	fall	year	reclat	reclong	Before 1970
id									
10036	Enigma	Valid	H4	94.0	Found	1967.0	31.33333	-82.31667	True
10037	Enon	Valid	Iron, ungrouped	763.0	Found	1883.0	39.86667	-83.95000	True
10038	Enshi	Valid	H5	8000.0	Fell	1974.0	30.30000	109.50000	False
10039	Ensisheim	Valid	LL6	127000.0	Fell	1491.0	47.86667	7.35000	True

In [263...

```
# Bonus
error = meteorite['year'].isna().sum()
print("Data entry error is in: ", error)
```

Data entry error is in: 291

In [ ]: