In [9]:
```
import pandas as pd

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

In [46]:
```
mask = taxis.columns.str.contains('id$|store_and_fwd_flag', regex = True)
columns_to_drop = taxis.columns[mask]
columns_to_drop
```

Out[46]:
```
Index([], dtype='object')
```

In [47]:
```
taxis = taxis.drop(columns = columns_to_drop)
taxis.head()
```

Out[47]:

	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	

Renaming Columns

In [11]:
```
taxis = taxis.rename(
    columns = {
        'tpep_pickup_datetime': 'pickup',
        'tpep_dropoff_datetime': 'dropoff'
    }

)
```

In [12]:
```
taxis.dtypes
```

Out[12]:
```
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 [19]:
```
taxis[['pickup', 'dropoff']] = taxis[['pickup', 'dropoff']].apply(pd.to_datetime)
taxis.dtypes
```

Out[19]:
```
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 [28]:
```
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['elapsed_time'] = taxis['dropoff']- taxis['pickup'] taxis

In [29]:
```
taxis.dtypes
```

Out[29]:
```
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 [36]:
```
taxis.sort_values(['trip_distance', 'fees'], ascending = [False, True]).head()
```

Out[36]:

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvem
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 [35]:
```
taxis.head()
```

Out[35]:

	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	

In [37]:
```
taxis.nlargest(3, 'elapsed_time')
```

Out[37]:

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvem
7576	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	
6902	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	
4975	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 [40]:
```
taxis.nlargest(5, 'fees') # for finding the largest rows, '.nsmallest()
```

Out[40]:

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvem
449	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	
8898	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	
3354	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	
9758	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	
3486	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 [42]:
```
# Loading the data
meteorite = pd.read_csv('Meteorite_Landings.csv')
```

In [43]:
```
# Renaming the mass (g) column
meteorite = meteorite.rename(
    columns = {
        'mass (g)': 'mass'
    }
)
meteorite
```

Out[43]:

	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 [61]:
```
# Drop latitude and longitude
mask = meteorite.columns.str.contains('GeoLocation', regex = True)
columns_to_drop = meteorite.columns[mask]
columns_to_drop
```

Out[61]:
```
Index([], dtype='object')
```

In [62]:
```
meteorite = meteorite.drop(columns = columns_to_drop)
meteorite.head()
```

Out[62]:

	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 [63]:
```
# Sort the result by mass in descending
sort = meteorite.sort_values(['mass'], ascending = [False])
sort
```

Out[63]:

	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

In [ ]: