In [6]:
```
import pandas as pd
```

In [3]:
```
meteorites = pd.read_csv("Meteorite_Landings.csv", nrows = 5)
meteorites
```

Out[3]:

	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	

In [4]:
```
meteorites.name
```

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

In [7]:
```
meteorites["name"]
```

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

In [5]:
```
meteorites.columns
```

Out[5]:
```
Index(['name', 'id', 'nametype', 'recclass', 'mass (g)', 'fall', 'year',
       'reclat', 'reclong', 'GeoLocation'],
      dtype='object')
```

In [6]:
```
meteorites.index
```

Out[6]:
```
RangeIndex(start=0, stop=5, step=1)
```

In [17]:
```
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 [20]:
```
payload[0]
```

Out[20]:
```
{'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 [22]:
```
df = pd.DataFrame(payload)
df.head(3)
```

Out[22]:

	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 [25]:
```
meteorites = pd.read_csv("Meteorite_Landings.csv")
meteorites.shape
```

Out[25]:
```
(45716, 10)
```

In [26]:
```
meteorites.columns
```

Out[26]:
```
Index(['name', 'id', 'nametype', 'recclass', 'mass (g)', 'fall', 'year',
       'reclat', 'reclong', 'GeoLocation'],
      dtype='object')
```

In [27]:
```
meteorites.dtypes
```

Out[27]:
```
name           object
id              int64
nametype       object
recclass       object
mass (g)      float64
fall           object
year           object
reclat        float64
reclong       float64
GeoLocation    object
dtype: object
```

In [30]:
```
meteorites.head(10)
```

Out[30]:

	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 [31]:
```
meteorites.tail(5)
```

Out[31]:

	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 [32]:
```
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 [43]:
```
meteorites[["name", "year"]]
```

Out[43]:

	name	year
0	Aachen	01/01/1880 12:00:00 AM
1	Aarhus	01/01/1951 12:00:00 AM
2	Abee	01/01/1952 12:00:00 AM
3	Acapulco	01/01/1976 12:00:00 AM
4	Achiras	01/01/1902 12:00:00 AM
...	...	...
45711	Zillah 002	01/01/1990 12:00:00 AM
45712	Zinder	01/01/1999 12:00:00 AM
45713	Zlin	01/01/1939 12:00:00 AM
45714	Zubkovsky	01/01/2003 12:00:00 AM
45715	Zulu Queen	01/01/1976 12:00:00 AM

45716 rows × 2 columns

In [44]:
```
meteorites[["name", "mass (g)"]]
```

Out[44]:

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

45716 rows × 2 columns

In [45]:
```
meteorites[100:104]
```

Out[45]:

	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 [49]:
```
meteorites.iloc[100:104, [0, 3, 4, 6]]
```

Out[49]:

	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 [50]:
```
meteorites.loc[100:104, "mass (g)": "year"]
```

Out[50]:

	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 [56]:
```
meteorites.iloc[-1, [-1]]
```

Out[56]:
```
GeoLocation    (33.98333, -115.68333)
Name: 45715, dtype: object
```

In [57]:
```
meteorites.iloc[:, [-1]]
```

Out[57]:

	GeoLocation
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)

45716 rows × 1 columns

In [60]:
```
(meteorites["mass (g)"] > 50) & (meteorites.fall == "Found")
```

Out[60]:
```
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 [61]:
```
meteorites[(meteorites["mass (g)"] > 1e6) & (meteorites.fall == "Fell")]
```

Out[61]:

	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 [64]:
```
meteorites.query("`mass (g)` > 1e6 and fall == 'Fell'")
```

Out[64]:

	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 [65]:
```
meteorites.value_counts()
```

Out[65]:
```
name                          id     nametype  recclass  mass (g)  fall   year
reclat      reclong     GeoLocation
Aachen                        1      Valid     L5        21.00     Fell   01/01/188
0 12:00:00 AM   50.775000  6.083330    (50.775, 6.08333)          1
Queen Alexandra Range 90215   19020  Valid     L5        358.90    Found  01/01/199
0 12:00:00 AM  -84.606700  162.167080  (-84.6067, 162.16708)      1
Queen Alexandra Range 90217   19022  Valid     L5        327.70    Found  01/01/199
0 12:00:00 AM  -84.595030  162.319900  (-84.59503, 162.3199)      1
Queen Alexandra Range 90218   19023  Valid     L5        926.50    Found  01/01/199
0 12:00:00 AM  -84.610170  162.153880  (-84.61017, 162.15388)     1
Queen Alexandra Range 90219   19024  Valid     L5        316.00    Found  01/01/199
0 12:00:00 AM  -84.611160  162.145660  (-84.61116, 162.14566)     1

..
Grove Mountains 022004        47902  Valid     H5        0.59      Found  01/01/200
3 12:00:00 AM  -72.776389  75.348333   (-72.776389, 75.348333)    1
Grove Mountains 022005        47903  Valid     L6        8.93      Found  01/01/200
3 12:00:00 AM  -72.776389  75.348611   (-72.776389, 75.348611)    1
Grove Mountains 022006        49930  Valid     L6        1.82      Found  01/01/200
3 12:00:00 AM  -72.776390  75.348890   (-72.77639, 75.34889)      1
Grove Mountains 022011        47904  Valid     L5        2.63      Found  01/01/200
3 12:00:00 AM  -72.780556  75.345278   (-72.780556, 75.345278)    1
Święcany                      47342  Valid     L/LL5     8.00      Found  01/01/200
4 12:00:00 AM   49.791390  21.257780   (49.79139, 21.25778)       1
Name: count, Length: 38115, dtype: int64
```

In [68]:
```
meteorites.fall.value_counts()
```

Out[68]:
```
fall
Found    44609
Fell      1107
Name: count, dtype: int64
```

In [66]:
```
meteorites.value_counts(subset=["nametype", "fall"], normalize = True)
```

Out[66]:
```
nametype  fall
Valid     Found    0.974145
          Fell     0.024215
Relict    Found    0.001641
Name: proportion, dtype: float64
```

In [82]:
```
meteorites["mass (g)"].mean()
```

Out[82]:
```
13278.078548601512
```

In [72]:
```
meteorites["mass (g)"].quantile([0.01, 0.05, 0.5, 0.95, 0.99])
```

Out[72]:
```
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 [76]:
```
meteorites["mass (g)"].median()
```

Out[76]: 32.6

In [77]:
```
meteorites["mass (g)"].max()
```

Out[77]: 60000000.0

In [81]:
```
meteorites.loc[meteorites["mass (g)"].idxmax()]
```

Out[81]:
```
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 [83]:
```
meteorites.recclass.nunique()
```

Out[83]: 466

In [84]:
```
meteorites.recclass.unique()
```

Out[84]:
```
array(['L5', 'H6', 'EH4', 'Acapulcoite', 'L6', 'LL3-6', 'H5', 'L',
       'Diogenite-pm', 'Unknown', 'H4', 'H', 'Iron, IVA', 'CR2-an', 'LL5',
       'CI1', 'L/LL4', 'Eucrite-mmict', 'CV3', 'Ureilite-an',
       'Stone-uncl', 'L3', 'Angrite', 'LL6', 'L4', 'Aubrite',
       'Iron, IIAB', 'Iron, IAB-sLL', 'Iron, ungrouped', 'CM2', 'OC',
       'Mesosiderite-A1', 'LL4', 'C2-ung', 'LL3.8', 'Howardite',
       'Eucrite-pmict', 'Diogenite', 'LL3.15', 'LL3.9', 'Iron, IAB-MG',
       'H/L3.9', 'Iron?', 'Eucrite', 'H4-an', 'L/LL6', 'Iron, IIIAB',
       'H/L4', 'H4-5', 'L3.7', 'LL3.4', 'Martian (chassignite)', 'EL6',
       'H3.8', 'H3-5', 'H5-6', 'Mesosiderite', 'H5-7', 'L3-6', 'H4-6',
       'Ureilite', 'Iron, IID', 'Mesosiderite-A3/4', 'CO3.3', 'H3',
       'EH3/4-an', 'Iron, IIE', 'L/LL5', 'H3.7', 'CBa', 'H4/5', 'H3/4',
       'H?', 'H3-6', 'L3.4', 'Iron, IAB-sHL', 'L3.7-6', 'EH7-an', 'Iron',
       'CR2', 'CO3.2', 'K3', 'L5/6', 'CK4', 'Iron, IIE-an', 'L3.6',
       'LL3.2', 'Pallasite', 'CO3.5', 'Lodranite', 'Mesosiderite-A3',
       'L3-4', 'H5/6', 'Pallasite, PMG', 'Eucrite-cm', 'L5-6', 'CO3.6',
       'Martian (nakhlite)', 'LL3.6', 'C3-ung', 'H3-4', 'CO3.4', 'EH3',
       'Iron, IAB-ung', 'Winonaite', 'LL', 'Eucrite-br', 'Iron, IIF',
       'R3.8-6', 'L4-6', 'EH5', 'LL3.00', 'H3.4', 'Martian (shergottite)',
       'Achondrite-ung', 'LL3.3', 'C', 'H/L3.6', 'Iron, IIIAB-an', 'LL7',
       'Mesosiderite-B2', 'LL4-6', 'CO3.7', 'L/LL6-an',
       'Iron, IAB complex', 'Pallasite, PMG-an', 'H3.9/4', 'L3.8',
       'LL5-6', 'LL3.8-6', 'L3.9', 'L4-5', 'L3-5', 'LL4/5', 'L4/5',
       'H3.9', 'H3.6-6', 'H3.8-5', 'H3.8/4', 'H3.9-5', 'CH3', 'R3.8-5',
       'L3.9/4', 'E4', 'CO3', 'Chondrite-ung', 'H~5', 'H~6', 'L/LL3.10',
       'EL5', 'LL3', 'L~6', 'L~3', 'H~4', 'L(LL)3.5-3.7', 'Iron, IIIE-an',
       'H3.6', 'L3.4-3.7', 'L3.5', 'CM1/2', 'Martian (OPX)', 'Brachinite',
       'LL7(?)', 'LL6(?)', 'Eucrite-Mg rich', 'H3.5-4', 'EL3', 'R3.6',
       'H3.5', 'CM1', 'L/LL3', 'H7', 'L(?)3', 'L3.2', 'L3.7-3.9',
       'Mesosiderite-B1', 'Eucrite-unbr', 'LL3.7', 'CO3.0', 'LL3.5',
       'L3.7-4', 'CV3-an', 'Lunar (anorth)', 'L3.3', 'Iron, IAB-sLM',
       'Lunar', 'Iron, IC', 'Iron, IID-an', 'Iron, IIIE', 'Iron, IVA-an',
       'CK6', 'L3.1', 'CK5', 'H3.3', 'H3.7-6', 'E6', 'H3.0', 'H3.1',
       'L3.0', 'L/LL3.4', 'C6', 'LL3.0', 'Lunar (gabbro)', 'R4', 'C4',
       'Iron, IIG', 'Iron, IIC', 'C1-ung', 'H5-an', 'EH4/5', 'Iron, IIIF',
       'R3-6', 'Mesosiderite-B4', 'L6/7', 'Relict H', 'L-imp melt', 'CK3',
       'H3-an', 'Iron, IVB', 'R3.8', 'L~5', 'Mesosiderite-an',
       'Mesosiderite-A2', 'Pallasite, PES', 'C4-ung', 'Iron, IAB?',
       'Mesosiderite-A', 'R3.5-6', 'H3.9-6', 'Ureilite-pmict', 'LL~6',
       'CK4/5', 'EL4', 'Lunar (feldsp. breccia)', 'L3.9-6', 'H-an',
       'L/LL3-6', 'L/LL3-5', 'H/L3.5', 'H/L3', 'R3-4', 'CK3-an', 'LL4-5',
       'H/L6', 'L3/4', 'H-imp melt', 'CR', 'Chondrite-fusion crust',
       'Iron, IAB-sLH', 'H(L)3-an', 'L(LL)3', 'H(L)3', 'R3', 'L7',
       'CM-an', 'L/LL~6', 'L/LL~5', 'L~4', 'L/LL~4', 'LL(L)3', 'H3.2',
       'L-melt breccia', 'H6-melt breccia', 'H5-melt breccia',
       'H-melt rock', 'Eucrite-an', 'Lunar (bas/anor)', 'LL5/6', 'LL3/4',
       'H3.4/3.5', 'Lunar (basalt)', 'H/L5', 'H(5?)', 'LL-imp melt',
       'Mesosiderite?', 'H~4/5', 'L6-melt breccia', 'L3.5-3.7',
       'Iron, IIAB-an', 'L3.3-3.7', 'L3.2-3.6', 'L3.3-3.6',
       'Acapulcoite/Lodranite', 'Mesosiderite-B', 'CK5/6', 'L3.05', 'C2',
       'C4/5', 'L/LL3.2', 'Iron, IIIAB?', 'L3.5-5', 'L/LL(?)3', 'H4(?)',
       'Iron, IAB-sHH', 'Relict iron', 'EL4/5', 'L5-7', 'Diogenite-an',
       'L-melt rock', 'CR1', 'H5 ', 'L5 ', 'H4 ', 'L4 ', 'E', 'L6 ',
       'H3 ', 'LL6 ', 'H-metal', 'H6 ', 'L-metal', 'Relict OC', 'EH',
       'Mesosiderite-A4', 'L/LL5/6', 'H3.8-4', 'CBb', 'EL6/7', 'EL7',
       'CH/CBb', 'CO3.8', 'H/L~4', 'Mesosiderite-C2', 'R5', 'H4/6',
```

```
'H3.7-5', 'LL3.7-6', 'H3.7/3.8', 'L3.7/3.8', 'EH-imp melt', 'R',
'Fusion crust', 'Aubrite-an', 'R6', 'LL-melt rock', 'L3.5-3.9',
'L3.2-3.5', 'L3.3-3.5', 'L3.0-3.7', 'E3-an', 'K', 'E3',
'Acapulcoite/lodranite', 'CK4-an', 'L(LL)3.05', 'L3.10', 'CB',
'Diogenite-olivine', 'EL-melt rock', 'EH6', 'Pallasite, ungrouped',
'L/LL4/5', 'L3.8-an', 'Iron, IAB-an', 'C5/6-ung', 'CV2',
'Iron, IC-an', 'Lunar (bas. breccia)', 'L3.8-6', 'R3/4', 'R3.9',
'CK', 'LL3.10', 'R4/5', 'L3.8-5', 'Mesosiderite-C', 'Enst achon',
'H/L3-4', 'L(H)3', 'LL6/7', 'LL3.1', 'OC3', 'R3.7', 'CO3 ', 'CH3 ',
'LL~4', 'LL~4/5', 'L(LL)~4', 'H3.05', 'H3.10',
'Impact melt breccia', 'LL3-5', 'H/L3.7', 'LL3-4', 'CK3/4',
'Martian', 'CO3.1', 'Lunar (bas/gab brec)', 'Achondrite-prim',
'LL<3.5', 'CK3.8', 'L/LL-melt rock', 'H6/7', 'EL6 ',
'Iron, IAB-sHL-an', 'CM2-an', 'R3-5', 'L4-melt rock',
'L6-melt rock', 'H/L4/5', 'EL3/4', 'H/L6-melt rock',
'Enst achon-ung', 'L3-7', 'R3.4', 'LL3.05', 'LL4/6', 'LL3.8-4',
'H3.15', 'C3.0-ung', 'LL-melt breccia', 'LL6-melt breccia',
'L5-melt breccia', 'LL(L)3.1', 'LL6-an', 'L4-melt breccia',
'Howardite-an', 'H4-melt breccia', 'Martian (basaltic breccia)',
'L3-melt breccia', 'L~4-6', 'LL~5', 'R3.5-4', 'CR7',
'H-melt breccia', 'Lunar (norite)', 'L3.00', 'H3.0-3.4', 'L/LL4-6',
'CM', 'EH7', 'L4-an', 'E-an', 'H3.8/3.9', 'L3.9-5', 'H3.8-6',
'H3.4-5', 'L3.0-3.9', 'L3.5-3.8', 'H3.2-3.7', 'L3.6-4',
'Iron, IIE?', 'C3/4-ung', 'L/LL3.5', 'L/LL3.6/3.7', 'H/L4-5',
'LL~3', 'Pallasite?', 'LL5-7', 'LL3.9/4', 'H3.8-an', 'CR-an',
'L/LL5-6', 'L(LL)5', 'L(LL)6', 'LL3.1-3.5', 'E5', 'Lodranite-an',
'H3.2-6', 'H(?)4', 'E5-an', 'H3.2-an', 'EH6-an', 'Stone-ung',
'C1/2-ung', 'L/LL'], dtype=object)
```

In [86]:
```
meteorites.describe(include = "all")
```

Out[86]:

	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).

1. Create a DataFrame by reading in the 2019_Yellow_Taxi_Trip_Data.csv file. Examine the first 5 rows.

In [61]:
```
data = pd.read_csv("2019_Yellow_Taxi_Trip_Data.csv")
dframe = pd.DataFrame(data)
dframe.head()
```

Out[61]:

	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

2. Find the dimensions (number of rows and number of columns) in the data.

In [62]:
```
dframe.shape
```

Out[62]:
```
(10000, 18)
```

The rows is 10000. The columns is 18.

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.

In [63]:
```
result = dframe.iloc[:, [4, 10, 13, 14, 16]]
result.describe()
```

Out[63]:

	trip_distance	fare_amount	tip_amount	tolls_amount	total_amount
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	3.015250	15.106313	2.634494	0.623447	22.564659
std	4.148063	13.954762	3.409800	6.437507	19.209255
min	0.000000	-52.000000	0.000000	-6.120000	-65.920000
25%	0.920000	7.000000	0.000000	0.000000	12.375000
50%	1.500000	10.000000	2.000000	0.000000	16.300000
75%	2.760000	16.000000	3.250000	0.000000	22.880000
max	38.110000	176.000000	43.000000	612.000000	671.800000

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

In [64]:
```
result.loc[result["trip_distance"].idxmax()]
```

Out[64]:
```
trip_distance     38.11
fare_amount      176.00
tip_amount        18.29
tolls_amount       6.12
total_amount     201.21
Name: 8338, dtype: float64
```

Reflection: In this activity, I had a hard time on the last part as I get confused on how would I isolate the fare_amount, tip_amount, tolls_amount, and total_amount for the longest trip by distance. All throughout the activity, I was able to apply all what I have understood on the lecture.

In [107…
```
taxis = pd.read_csv("2019_Yellow_Taxi_Trip_Data.csv")
```

In [108…
```
mask = taxis.columns.str.contains("id$|store_and_fwd_flag", regex = True)
columns_to_drop = taxis.columns[mask]
columns_to_drop
```

Out[108…
```
Index(['vendorid', 'ratecodeid', 'store_and_fwd_flag', 'pulocationid',
       'dolocationid'],
      dtype='object')
```

In [109…
```
taxis = taxis.drop(columns = columns_to_drop)
taxis.head()
```

Out[109…

	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	payment_t
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	

In [110…

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

Out[110…

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

In [111…

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

Out[111…

```
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
```

In [112…

```
taxis = taxis.assign(
    elapsed_time = lambda x: x.dropoff - x.pickup,
    cost_before_tip = lambda x: x.total_amount - x.tip_amount,
    tip_pct = lambda x: x.tip_amount / x.cost_before_tip,
    fees = lambda x: x.cost_before_tip - x.fare_amount,
    avg_speed = lambda x: x.trip_distance.div(
        x.elapsed_time.dt.total_seconds()/60/60
```

```
    )
)
#It create columns with computed values

taxis.dtypes
```

Out[112…

```
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 [113…

```
taxis.head(2)
```

Out[113…

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	m
0	2019-10-23 16:39:42	2019-10-23 17:14:10	1	7.93	1	29.5	1.0	
1	2019-10-23 16:32:08	2019-10-23 16:45:26	1	2.00	1	10.5	1.0	

In [114…

```
taxis.sort_values(["passenger_count", "pickup"], ascending = [False, True]).head()
```

Out[114…

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra
5997	2019-10-23 15:55:19	2019-10-23 16:08:25	6	1.58	2	10.0	1.0
443	2019-10-23 15:56:59	2019-10-23 16:04:33	6	1.46	2	7.5	1.0
8722	2019-10-23 15:57:33	2019-10-23 16:03:34	6	0.62	1	5.5	1.0
4198	2019-10-23 15:57:38	2019-10-23 16:05:07	6	1.18	1	7.0	1.0
8238	2019-10-23 15:58:31	2019-10-23 16:29:29	6	3.23	2	19.5	1.0

In [115…

```
taxis.nlargest(3, "elapsed_time")
#It shows the largest value in elapsed_time
```

Out[115…

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra
7576	2019-10-23 16:52:51	2019-10-24 16:51:44	1	3.75	1	17.5	1.0
6902	2019-10-23 16:51:42	2019-10-24 16:50:22	1	11.19	2	39.5	1.0
4975	2019-10-23 16:18:51	2019-10-24 16:17:30	1	0.70	2	7.0	1.0

In [84]:

```
taxis.nlargest(3, "fare_amount")
```

Out[84]:

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra
8338	2019-10-23 16:50:53	2019-10-24 15:32:55	1	38.11	1	176.00	0.0
853	2019-10-23 16:07:39	2019-10-23 17:37:05	3	19.09	2	160.00	0.0
4714	2019-10-23 16:33:17	2019-10-23 17:56:49	2	26.30	1	111.75	0.0

Exercise (Part 2)

Read in the meteorite data from the Meteorite_Landings.csv file, rename the mass (g) column to mass, and drop all the latitude and longtitude columns. Sort the result by mass in descending order.

In [101…

```
# Import meteorite data from the Meteorite_Landings.csv file
meteorite1 = pd.read_csv("Meteorite_Landings.csv")
meteorite1
```

Out[101…

	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 [102…

```
# Rename the mass (g) column to mass
meteorite1 = meteorite1.rename(
    columns={
        'mass (g)': 'mass',
    }
)
meteorite1
```

Out[102…

	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 [103…

```
# Drop all the latitude and longtitude columns
meteorite1 = meteorite1.drop(columns = ["reclat", "reclong"])
meteorite1
```

Out[103…

	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 [104…

```
meteorite1 = meteorite1.sort_values("mass", ascending = False)
meteorite1
```

Out[104…

	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 [116…

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

Out[116...

pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta
2019-10-23 16:39:42	2019-10-23 17:14:10	1	7.93	1	29.5	1.0	
2019-10-23 16:32:08	2019-10-23 16:45:26	1	2.00	1	10.5	1.0	
2019-10-23 16:08:44	2019-10-23 16:21:11	1	1.36	1	9.5	1.0	

In [129...

```
taxis.sort_index(axis = 0)
# This sort the by index, but axis = 1 is sorted by columns
```

Out[129...

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

10000 rows × 17 columns

In [139...

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

Out[139…

pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta
2019-10-23 07:48:58	2019-10-23 07:52:09	1	0.67	2	4.5	1.0	
2019-10-23 08:02:09	2019-10-24 07:42:32	1	8.38	1	32.0	1.0	
2019-10-23 08:18:47	2019-10-23 08:36:05	1	2.39	2	12.5	1.0	

In [140…

```
taxis["2019-10-23": "2019-10-23"]
```

Out[140…

pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta
2019-10-23 07:05:34	2019-10-23 08:03:16	3	14.68	1	50.0	1.0	
2019-10-23 07:48:58	2019-10-23 07:52:09	1	0.67	2	4.5	1.0	
2019-10-23 08:02:09	2019-10-24 07:42:32	1	8.38	1	32.0	1.0	
2019-10-23 08:18:47	2019-10-23 08:36:05	1	2.39	2	12.5	1.0	
2019-10-23 09:27:16	2019-10-23 09:33:13	2	1.11	2	6.0	1.0	
…	…	…	…	…	…	…	
2019-10-23 17:59:53	2019-10-23 18:12:56	2	1.60	1	10.0	1.0	
2019-10-23 17:59:53	2019-10-23 18:19:12	1	2.39	2	14.0	1.0	
2019-10-23 18:00:03	2019-10-23 18:04:56	1	0.94	2	5.5	1.0	
2019-10-23 18:01:21	2019-10-23 18:08:00	5	1.25	1	6.5	1.0	
2019-10-23 18:03:03	2019-10-23 18:10:45	1	0.76	1	6.5	1.0	

9993 rows × 17 columns

In [148…

```
taxis.loc["2019-10-23 08"]
```

Out[148...

pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	mta
2019-10-23 08:02:09	2019-10-24 07:42:32	1	8.38	1	32.0	1.0	
2019-10-23 08:18:47	2019-10-23 08:36:05	1	2.39	2	12.5	1.0	

In [149...

```
taxis = taxis.reset_index()
taxis.head()
# This the index
```

Out[149...

	pickup	dropoff	passenger_count	trip_distance	payment_type	fare_amount	extra	m
0	2019-10-23 07:05:34	2019-10-23 08:03:16	3	14.68	1	50.0	1.0	
1	2019-10-23 07:48:58	2019-10-23 07:52:09	1	0.67	2	4.5	1.0	
2	2019-10-23 08:02:09	2019-10-24 07:42:32	1	8.38	1	32.0	1.0	
3	2019-10-23 08:18:47	2019-10-23 08:36:05	1	2.39	2	12.5	1.0	
4	2019-10-23 09:27:16	2019-10-23 09:33:13	2	1.11	2	6.0	1.0	

Exercise (Part 3)

In [235...

```
meteorite2 = pd.read_csv("Meteorite_Landings.csv")
meteorite2
```

Out[235...

	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 [236...

```
# Update the year column to only contain the year# Import meteorite data from the M
meteorite2_1 = lambda x: x.str[6:10]
meteorite2["year"] = meteorite2_1(meteorite2["year"])
meteorite2
```

Out[236...

	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 [238...

```
# Convert it to a numeric data type
meteorite2["year"] = meteorite2["year"].apply(pd.to_numeric)
meteorite2.dtypes
```

Out[238...

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

In [242...

```
# Create a new column indicating whether the meteorite was observed falling before
meteorite2["Observed_Falling"] = (meteorite2.year < 1970.0) & (meteorite2.fall == "
meteorite2
```

Out[242...

	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 [245...

```
# Set the index to the id column and extract all the rows with IDs between 10036 an
meteorite2.set_index("id")
```

Out[245...

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

45716 rows × 10 columns

In [246...

```
meteorite2.sort_index()
```

Out[246...

	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 [247...

```
meteorite2.iloc[10036: 10040]
```

Out[247...

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong	Ge
10036	Elephant Moraine 90022	8432	Valid	CK5	15.5	Found	1990.0	-76.28573	156.45721	(1
10037	Elephant Moraine 90023	8433	Valid	CK5	31.5	Found	1990.0	-76.27507	156.41038	(1
10038	Elephant Moraine 90024	8434	Valid	Eucrite-br	22.8	Found	1990.0	-76.28843	156.47872	(1
10039	Elephant Moraine 90025	8435	Valid	CK5	45.8	Found	1990.0	-76.28200	156.39926	1

Exercise (Part 4)

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

In [5]:
```
data = pd.read_csv("Meteorite_Landings.csv")
meteorites = pd.DataFrame(data)
meteorites.head()
```

Out[5]:

	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

1. Using the meteorite data from the Meteorite_Landings.csv file, create a pivot table that shows both the number of meteorites and the 95th percentile of meteorite mass for

those that were found versus observed falling per year from 2005 through 2009 (inclusive). Hint: Be sure to convert the year column to a number as we did in the previous exercise.

In [7]:
```
filter = lambda x: x.str[6:10]
meteorites["year"] = filter(meteorites["year"])
meteorites["year"] = meteorites["year"].apply(pd.to_numeric)
meteorites.dtypes
```

Out[7]:
```
name            object
id               int64
nametype        object
recclass        object
mass (g)       float64
fall            object
year           float64
reclat         float64
reclong        float64
GeoLocation     object
dtype: object
```

In [9]:
```
meteorites_filtered = meteorites[(meteorites["year"] >= 2005) & (meteorites["year"]
meteorites_filtered = meteorites_filtered.set_index("year")
```

In [11]:
```
meteorites_Fell = meteorites_filtered[(meteorites_filtered["fall"] == "Fell")]
meteorites_Fell.iloc[:, 4].quantile([0.95])
```

Out[11]:
```
0.95    100000.0
Name: mass (g), dtype: float64
```

In [13]:
```
meteorites_Found = meteorites_filtered[(meteorites_filtered["fall"] == "Found")]
meteorites_Found.iloc[:, 4].quantile([0.95])
```

Out[13]:
```
0.95    1841.64
Name: mass (g), dtype: float64
```

In [15]:
```
Number_1_1 = meteorites_filtered.groupby(["year", "fall"])[["mass (g)"]].quantile(0
Number_1_1
```

Out[15]:

		mass (g)
year	fall	
2005.0	Found	4500.00
2006.0	Fell	25008.00
	Found	1600.50
2007.0	Fell	89675.00
	Found	1126.90
2008.0	Fell	106000.00
	Found	2274.80
2009.0	Fell	8333.40
	Found	1397.25

In [17]:
```
Number_1_2 = meteorites_filtered.groupby(["year", "fall"])[["name"]].count()
Number_1_2
```

Out[17]:

		name
year	fall	
2005.0	Found	875
2006.0	Fell	5
	Found	2451
2007.0	Fell	8
	Found	1181
2008.0	Fell	9
	Found	948
2009.0	Fell	5
	Found	1492

In [19]:
```
Number_1_1["95th Percentile"] = Number_1_1
Number_1_1["Count"] = Number_1_2
Number_1_1
```

Out[19]:

year	fall	mass (g)	95th Percentile	Count
2005.0	**Found**	4500.00	4500.00	875
2006.0	**Fell**	25008.00	25008.00	5
	Found	1600.50	1600.50	2451
2007.0	**Fell**	89675.00	89675.00	8
	Found	1126.90	1126.90	1181
2008.0	**Fell**	106000.00	106000.00	9
	Found	2274.80	2274.80	948
2009.0	**Fell**	8333.40	8333.40	5
	Found	1397.25	1397.25	1492

2. Using the meteorite data from the Meteorite_Landings.csv file, compare summary statistics of the mass column for the meteorites that were found versus observed falling.

In [21]:
```
Number_2 = meteorites_filtered.groupby("fall")["mass (g)"].describe()
Number_2
```

Out[21]:

fall	count	mean	std	min	25%	50%	75%	max
Fell	27.0	19029.665185	34081.623779	18.41	410.0	3950.0	8206.5	110000.0
Found	6945.0	1573.986245	42020.893987	0.00	7.5	34.5	197.0	3000000.0

Exercise (Part 5)

In [25]:
```
data1 = pd.read_csv("2019_Yellow_Taxi_Trip_Data.csv")
taxi = pd.DataFrame(data1)
taxi.head()
```

Out[25]:

	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

1. Using the taxi trip data in the 2019_Yellow_Taxi_Trip_Data.csv file, resample the data to an hourly frequency based on the dropoff time. Calculate the total trip_distance, fare_amount, tolls_amount, and tip_amount, then find the 5 hours with the most tips.

In [27]:
```
# resample the data to an hourly frequency based on the dropoff time
taxi['tpep_dropoff_datetime'] = pd.to_datetime(taxi['tpep_dropoff_datetime'])
taxi.set_index("tpep_dropoff_datetime", inplace = True)
taxi.index = pd.to_datetime(taxi.index)
taxi['Hour'] = taxi.index.hour

# Calculate the total trip_distance, fare_amount, tolls_amount, and tip_amount, the
taxi = taxi.groupby('Hour')[['trip_distance', 'fare_amount', 'tolls_amount', 'tip_a
taxi = taxi["tip_amount"].nlargest(5)
taxi
```

Out[27]:
```
Hour
16    12249.32
17    12044.03
18     1907.64
15       75.10
19       25.74
Name: tip_amount, dtype: float64
```

In [ ]: