Exercise Part 4:

- 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.
- 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.

import pandas as pd
Read the meteorite data from the CSV file
meteorite = pd.read_csv('Meteorite_Landings.csv')
meteorite.head(5)



Next steps:

View recommended plots

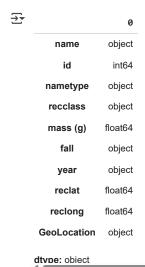
New interactive sheet

meteorite['year'] = meteorite['year'].str.slice(6,11) #Convert 'year' column to a numeric format by extracting year from the string meteorite.head(3)

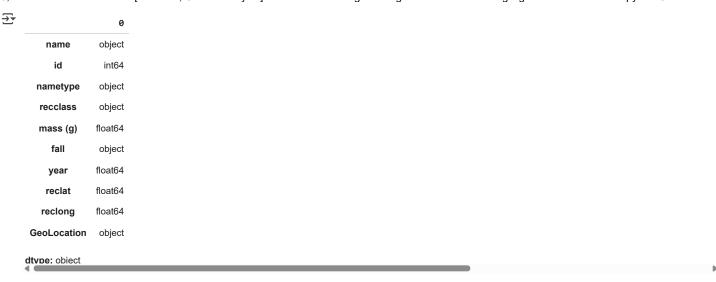


Next steps: (View recommended plots) (New interactive sheet

meteorite.dtypes



meteorite['year'] = pd.to_numeric(meteorite['year']) #Convert the 'year' column to numeric
meteorite.dtypes # Check data types after conversion



meteorite = meteorite[(meteorite['year'] >= 2005) & (meteorite['year'] <= 2009)] # Filter the data for the years 2005 to 2009 (inclusive)
years = meteorite['year'].unique() # Display unique years to confirm filtering
print(years)</pre>

→ [2008. 2009. 2006. 2007. 2005.]

meteorite.set_index('year') #sets index as year

	name	id	nametype	recclass	mass (g)	fall	reclat	reclong	GeoLocation
year									
2008.0	Almahata Sitta	48915	Valid	Ureilite-an	3950.000	Fell	20.74575	32.41275	(20.74575, 32.41275)
2009.0	Ash Creek	48954	Valid	L6	9500.000	Fell	31.80500	-97.01000	(31.805, -97.01)
2006.0	Bassikounou	44876	Valid	H5	29560.000	Fell	15.78333	-5.90000	(15.78333, -5.9)
2008.0	Berduc	48975	Valid	L6	270.000	Fell	-31.91000	-58.32833	(-31.91, -58.32833)
2007.0	Bunburra Rockhole	48653	Valid	Eucrite	324.000	Fell	-31.35000	129.19000	(-31.35, 129.19)
2008.0	Yabrin 003	48974	Valid	Acapulcoite	21.048	Found	23.31522	48.62988	(23.31522, 48.62988)
2006.0	Yaringie Hill	48950	Valid	H5	5750.000	Found	-32.08287	135.64985	(-32.08287, 135.64985)
2009.0	Yarle Lakes 004	52945	Valid	CK4	4.600	Found	-30.50000	131.46667	(-30.5, 131.46667)
2007.0	Yelland Dry Lake	52641	Valid	H4	76000.000	Found	39.35067	-114.40783	(39.35067, -114.40783)
2006.0	Youxi	EE702	Valid	Mesosiderite-C	218000.000	Found	26.06000	118.01000	(26.06, 118.01)

#1.1

 $Number_1 = meteorite.groupby(["year", "fall"])[["mass (g)"]].quantile(0.95) \# computes the quantile of year and fall by mass (g) \\ Number_1$



Exercise Part 5:

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.

```
import pandas as pd
# Read the Yellow Taxi from the CSV file
Taxi = pd.read_csv('2019_Yellow_Taxi_Trip_Data.csv')
Taxi.head(5)
```

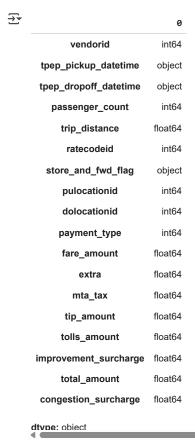
_	vendorid	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	ratecodeid	store_and_fwd_flag	pulocationid (
	2	2019-10- 23T16:39:42.000	2019-10- 23T17:14:10.000	1	7.93	1	N	138
,	I 1	2019-10- 23T16:32:08.000	2019-10- 23T16:45:26.000	1	2.00	1	N	11
2	2 2	2019-10- 23T16:08:44.000	2019-10-23T16:21:11.000	1	1.36	1	N	163
;	3 2	2019-10- 23T16:22:44.000	2019-10- 23T16:43:26.000	1	1.00	1	N	170
4	1 2	2019-10- 23T16:45:11.000	2019-10- 23T16:58:49.000	1	1.96	1	N	163

Next steps:

View recommended plots

New interactive sheet

Taxi.dtypes #check the datatype it must be in date_time so that we can use the hour



Taxi['tpep_dropoff_datetime'] = pd.to_datetime(Taxi['tpep_dropoff_datetime'])
Taxi.dtypes #then convert it

