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)
```

| | name | id | nametype | recclass | mass (g) | 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) |

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)
```

| | name | id | nametype | recclass | mass (g) | fall | year | reclat | reclong | GeoLocation |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Aachen | 1 | Valid | L5 | 21.0 | Fell | 1880 | 50.77500 | 6.08333 | (50.775, 6.08333) |
| 1 | Aarhus | 2 | Valid | H6 | 720.0 | Fell | 1951 | 56.18333 | 10.23333 | (56.18333, 10.23333) |
| 2 | Abee | 6 | Valid | EH4 | 107000.0 | Fell | 1952 | 54.21667 | -113.00000 | (54.21667, -113.0) |

Next steps: View recommended plots New interactive sheet

```
meteorite.dtypes
```

| | 0 |
|---|---|
| name | object |
| id | int64 |
| nametype | object |
| recclass | object |
| mass (g) | float64 |
| fall | object |
| year | object |
| reclat | float64 |
| reclong | float64 |
| GeoLocation | object |

dtype: object

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

| | 0 |
|---|---|
| **name** | object |
| **id** | int64 |
| **nametype** | object |
| **recclass** | object |
| **mass (g)** | float64 |
| **fall** | object |
| **year** | float64 |
| **reclat** | float64 |
| **reclong** | float64 |
| **GeoLocation** | object |

**dtype:** object

```
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)
```

```
[2008. 2009. 2006. 2007. 2005.]
```

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

| year | name | id | nametype | recclass | mass (g) | fall | reclat | reclong | GeoLocation |
|---|---|---|---|---|---|---|---|---|---|
| **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 | 55793 | Valid | Mesosiderite-C | 218000.000 | Found | 26.06000 | 118.01000 | (26.06, 118.01) |

6974 rows × 9 columns

```
#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
```

| | | 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 |

Next steps: View recommended plots New interactive sheet

```
#1.2
counts = meteorite.groupby(['year','fall'])['name'].count()

Number_1['Count'] = counts # to view the count in a more organized way
Number_1
```

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

Next steps: View recommended plots New interactive sheet

```
#2

summary = meteorite.groupby('fall')['mass (g)'].describe() #summary statistics of both fell and found
summary
```

| 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 |

Next steps: View recommended plots New interactive sheet

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 |
|---|---|---|---|---|---|---|---|---|
| **0** | 2 | 2019-10-23T16:39:42.000 | 2019-10-23T17:14:10.000 | 1 | 7.93 | 1 | N | 138 |
| **1** | 1 | 2019-10-23T16:32:08.000 | 2019-10-23T16:45:26.000 | 1 | 2.00 | 1 | N | 11 |
| **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** | 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
```

| | 0 |
|---|---|
| **vendorid** | int64 |
| **tpep_pickup_datetime** | object |
| **tpep_dropoff_datetime** | 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

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

| | 0 |
|---|---|
| **vendorid** | int64 |
| **tpep_pickup_datetime** | object |
| **tpep_dropoff_datetime** | datetime64[ns] |
| **passenger_count** | int64 |
| **trip_distance** | float64 |
| **ratecodeid** | int64 |
| **store_and_fwd_flag** | object |
| **pulocationid** | int64 |
| **dolocationid** | int64 |

```
Taxi.set_index('tpep_dropoff_datetime', inplace = True)
```

**fare_amount** float64

```
Taxi.index = pd.to_datetime(Taxi.index)


Taxi['hour'] = Taxi.index.hour
Taxi = Taxi.groupby('hour')[['trip_distance','fare_amount','tolls_amount','tip_amount']].sum()
Taxi
```

**improvement_surcharge** float64

**total_amount** float64

**dtype:** object

| hour | trip_distance | fare_amount | tolls_amount | tip_amount |
|---|---|---|---|---|
| **0** | 8.62 | 41.50 | 0.00 | 7.14 |
| **7** | 14.58 | 62.50 | 0.00 | 5.50 |
| **8** | 17.07 | 172.90 | 0.00 | 4.00 |
| **9** | 14.78 | 100.00 | 0.00 | 0.00 |
| **12** | 4.83 | 26.00 | 0.00 | 4.56 |
| **13** | 6.98 | 35.50 | 0.00 | 4.45 |
| **14** | 37.84 | 155.00 | 6.12 | 17.46 |
| **15** | 68.95 | 461.50 | 6.12 | 75.10 |
| **16** | 10715.39 | 68013.76 | 699.04 | 12249.32 |
| **17** | 16060.31 | 70160.91 | 4044.04 | 12044.03 |
| **18** | 3104.56 | 11565.56 | 1454.67 | 1907.64 |
| **19** | 98.59 | 268.00 | 24.48 | 25.74 |

Next steps: View recommended plots | New interactive sheet

```
Taxi = Taxi['tip_amount'].nlargest(5)
Taxi
```

| hour | tip_amount |
|---|---|
| **16** | 12249.32 |
| **17** | 12044.03 |
| **18** | 1907.64 |
| **15** | 75.10 |
| **19** | 25.74 |

**dtype:** float64