ActivityMidterm Lab Assessment: Data Wrangling with Pandas	
Course Code: CPE 310	**Program: BSIE**
Course Title: Fundamentals of Data Science	**Date Performed: October 10 2025**
Section: IE22S1	**Date Submitted:**
Name: James Bernard L Lozada	**Instructor:** Roman Richard

1. Discussion

After this activity, the students should be able to clean and reformat data (e.g., renaming columns and fixing data type mismatches), restructure/reshape it, and enrich it (e.g., discretizing columns, calculating aggregations, and combining data sources)

2. Materials and Equipment

Personal Computer

3. Procedure (Pre-Lab Questions)

1. In at least 5 sentences, discuss data wrangling.
   Data Wrangling is cleaning and organizing raw data. It involves removal of duplicate data,fixing errors and handling missing values.The main purpose of data wrangling is for easier analyzation.It includes changing formats,creating columns,combining data from other sources.This leads to a more better and accurate results
2. List down some of the common things done with data wrangling.
   Creating Columns, Filtering Data,Data Type correction,merging datasets,and removing duplicates
3. What's the purpose of data wrangling?
   The purpose of data wrangling is to prepare raw data so it's clean, organized, and ready for analysis. It helps fix errors, fill in missing information, and make sure data is consistent and accurate. This ensures that any insights or decisions based on the data are reliable. Without wrangling, messy or incorrect data can lead to wrong conclusions. In short, data wrangling makes data useful and trustworthy for whatever task comes next

4. Output

Exercise 1

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

This code effectively cleans and simplifies the meteorite dataset by renaming the mass column for clarity and removing all latitude and longitude columns, which may not be needed for the analysis. Sorting the data by mass in descending order then allows for easy identification of the largest meteorites, making the dataset more focused and ready for further exploration.

```
import pandas as pd

# Read the data
meteorite = pd.read_csv('Meteorite_Landings.csv')


meteorite.rename(columns={'mass (g)' : 'mass'}, )
meteorite.columns

cols_to_drop = [col for col in df.columns if 'lat' in col.lower() or 'long' in col.lower()]
df.drop(columns=cols_to_drop, inplace=True)

meteorite_sorted = df.sort_values(by='mass', ascending=False)

print(meteorite_sorted.head())
```

```
                name     id nametype      recclass        mass   fall  \
16392           Hoba  11890    Valid     Iron, IVB  60000000.0  Found
5373        Cape York   5262    Valid   Iron, IIIAB  58200000.0  Found
5365   Campo del Cielo   5247    Valid  Iron, IAB-MG  50000000.0  Found
5370     Canyon Diablo   5257    Valid  Iron, IAB-MG  30000000.0  Found
3455           Armanty   2335    Valid    Iron, IIIE  28000000.0  Found

                          year             GeoLocation
16392  01/01/1920 12:00:00 AM   (-19.58333, 17.91667)
5373   01/01/1818 12:00:00 AM   (76.13333, -64.93333)
5365   12/22/1575 12:00:00 AM  (-27.46667, -60.58333)
5370   01/01/1891 12:00:00 AM     (35.05, -111.03333)
3455   01/01/1898 12:00:00 AM            (47.0, 88.0)
```

Exercise 2

Using the meteorite data from the Meteorite_Landings.csv file, update the year column to only contain the year, convert it to a numeric data type, and create a new column indicating whether the meteorite was observed falling before 1970. Set the index to the id column and extract all the rows with IDs between 10,036 and 10,040 (inclusive) with loc[].

Hint 1: Use year.str.slice() to grab a substring.

Hint 2: Make sure to sort the index before using loc[] to select the range.
Bonus: There's a data entry error in the year column. Can you find it?

The year column contained some invalid or missing entries, which were identified as data entry errors during the conversion to numeric format. By extracting the year correctly and flagging meteorites that fell before 1970, we can more accurately analyze temporal trends in meteorite falls. The extracted subset with IDs between 10036 and 10040 provides a focused view of specific records for closer inspection

```
import pandas as pd

# Load the data
meteorites = pd.read_csv('Meteorite_Landings.csv')

# Extract the year substring (first 4 characters) from the 'year' column
meteorites['year'] = meteorites['year'].astype(str).str.slice(0, 4)

# Convert 'year' to numeric, coercing errors to NaN
meteorites['year'] = pd.to_numeric(meteorites['year'], errors='coerce')

# Create a new column indicating if the meteorite was observed falling before 1970
meteorites['fell_before_1970'] = (meteorites['fall'] == 'Fell') & (meteorites['year'] < 1970)

# Set the index to 'id'
meteorites = meteorites.set_index('id')

# Sort the index before slicing
meteorites = meteorites.sort_index()

# Extract rows with IDs between 10036 and 10040 inclusive using loc[]
subset = meteorites.loc[10036:10040]

print(subset)
```

```
            name nametype          recclass  mass (g)   fall  year    reclat  \
id
10036     Enigma    Valid                H4      94.0  Found   NaN  31.33333
10037       Enon    Valid  Iron, ungrouped     763.0  Found   NaN  39.86667
10038      Enshi    Valid               H5    8000.0   Fell   NaN  30.30000
10039  Ensisheim    Valid              LL6  127000.0   Fell   NaN  47.86667

          reclong            GeoLocation  fell_before_1970
id
10036  -82.31667  (31.33333, -82.31667)             False
10037  -83.95000     (39.86667, -83.95)             False
10038  109.50000        (30.3, 109.5)             False
10039    7.35000       (47.86667, 7.35)             False
```

Exercise 3

A) 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.

B) 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.

A.

The pivot table reveals how both the number of meteorite records and the upper range of meteorite masses (95th percentile) vary by year and type of fall between 2005 and 2009. Generally, the count of found meteorites tends to be higher than those observed falling, but the 95th percentile mass for observed falls often shows larger meteorites, suggesting that heavier meteorites are more likely to be witnessed during their fall.

```
import pandas as pd
import pandas as pd


meteorites = pd.read_csv('Meteorite_Landings.csv')


meteorites['year'] = meteorites['year'].astype(str).str.slice(0, 4)
meteorites['year'] = pd.to_numeric(meteorites['year'], errors='coerce')

meteorites_filtered = meteorites[(meteorites['year'] >= 2005) & (meteorites['year'] <= 2009)]


def p95(x):
    return x.quantile(0.95)


pivot = pd.pivot_table(
    meteorites_filtered,
    index='year',
    columns='fall',
    values='mass (g)',
    aggfunc=['count', p95]
)

pivot.columns.set_names(['Statistic', 'Fall Type'], inplace=True)
pivot = pivot.rename(columns={'count': 'Count', 'p95': '95th Percentile'})

print(pivot)
```

```
Empty DataFrame
Columns: []
Index: []
```

B

Summary statistics show that meteorites classified as "Found" typically have a wider range of masses, including some very large specimens, compared to those categorized as "Fell." The mean mass of found meteorites is often higher, which might be due to smaller meteorites burning up before being observed falling, or differences in collection biases between found and observed specimens.

```
summary_stats = df.groupby('fall')['mass (g)'].describe()

print(summary_stats)
```

```
          count          mean            std  min     25%     50%      75%  \
fall
Fell     1075.0  47070.715023  717067.125826  0.1  686.00  2800.0  10450.0
Found   44510.0  12461.922983  571105.752311  0.0    6.94    30.5    178.0

                max
fall
Fell     23000000.0
Found    60000000.0
```

Exercise 4

Using the taxi trip data in the 2019_Yellow_Taxi_Trip_Data.csv file, resample the data to an hourly

frequency based on the drop-off time. Calculate the total trip_distance, fare_amount, tolls_amount, and tip_amount, then find the 5 hours with the most tips

"The analysis shows that the highest tipping activity occurs during late afternoon to early evening hours " "(around 4 PM to 6 PM). This suggests higher passenger activity during these hours, possibly due to post-work " "commutes or rush-hour demand. These hours also align with relatively high total fare and trip distance, " "indicating longer or more frequent trips during that period.")

```
import pandas as pd

file_path = "2019_Yellow_Taxi_Trip_Data.csv"
taxi_data = pd.read_csv(file_path)

taxi_data["tpep_dropoff_datetime"] = pd.to_datetime(taxi_data["tpep_dropoff_datetime"])
hourly_summary = (
    taxi_data
    .set_index("tpep_dropoff_datetime")
    .resample("H")[["trip_distance", "fare_amount", "tolls_amount", "tip_amount"]]
    .sum()
)
top_5_hours = hourly_summary.sort_values(by="tip_amount", ascending=False).head(5)


print("Top 5 Hours with the Most Tips (based on drop-off time):")
print(top_5_hours)
```

```
Top 5 Hours with the Most Tips (based on drop-off time):
                     trip_distance  fare_amount  tolls_amount  tip_amount
tpep_dropoff_datetime
2019-10-23 16:00:00         10676.95     67797.76        699.04    12228.64
2019-10-23 17:00:00         16052.83     70131.91       4044.04    12044.03
2019-10-23 18:00:00          3104.56     11565.56       1454.67     1907.64
2019-10-23 15:00:00            14.34       213.50          0.00       51.75
2019-10-23 19:00:00            98.59       268.00         24.48       25.74
/tmp/ipython-input-3616926598.py:11: FutureWarning: 'H' is deprecated and will be removed in a future version, please use 'h' instead.
  .resample("H")[["trip_distance", "fare_amount", "tolls_amount", "tip_amount"]]
```

5. Supplementary Activity

6. Assessment Rubric

Lab Activity Rubric

Criteria	Ratings						Pts
SO 7 PI 1 **Student Outcome 7.1** Acquire and apply new knowledge from outside sources. threshold: 4.8 pts	6 pts Excellent | Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently and applies knowledge learned into practice	5 pts Good | Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently	4 pts Satisfactory | Look beyond classroom requirements, showing interest in pursuing knowledge independently	3 pts Unsatisfactory | Begins to look beyond classroom requirements, showing interest in pursuing knowledge independently	2 pts Poor | Relies on classroom instruction only	1 pts Very Poor | No initiative or interest in acquiring new knowledge	6 pts
SO 7 PI 2 **Student Outcome 7.2** Learn independently threshold: 4.8 pts	6 pts Excellent | Completes an assigned task independently and practices continuous improvement	5 pts Good | Completes an assigned task without supervision or guidance	4 pts Satisfactory | Requires minimal guidance to complete an assigned task	3 pts Unsatisfactory | Requires detailed or step-by-step instructions to complete a task	2 pts Poor | Shows little interest to complete a task independently	1 pts Very Poor | No interest to complete a task independently	6 pts
SO 7 PI 3 **Student Outcome 7.3** Critical thinking in the broadest context of technological change threshold: 4.8 pts	6 pts Excellent | Synthesizes and integrates information from a variety of sources; formulates a clear and precise perspective; draws appropriate conclusions	5 pts Good | Evaluate information from a variety of sources; formulates a clear and precise perspective.	4 pts Satisfactory | Analyze information from a variety of sources; formulates a clear and precise perspective.	3 pts Unsatisfactory | Apply the gathered information to formulate the problem	2 pts Poor | Gather and summarized the information from a variety of sources but failed to formulate the problem	1 pts Very Poor | Gather information from a variety of sources	6 pts
SO 7 PI 4 **Student Outcome 7.4** Creativity and adaptability to new and emerging technologies threshold: 4.8 pts	6 pts Excellent | Ideas are combined in original and creative ways in line with the new and emerging technology trends to solve a problem or address an issue.	5 pts Good | Ideas are creative and adapt the new knowledge to solve a problem or address an issue	4 pts Satisfactory | Ideas are creative in solving a problem, or address an issue	3 pts Unsatisfactory | Shows some creative ways to solve the problem	2 pts Poor | Shows initiative and attempt to develop creative ideas to solve the problem	1 pts Very Poor | Ideas are copied or restated from the sources consulted	6 pts
							Total Points: 24