**Guided Data Assignment: Reading & Subsetting Data in Python**

**1. Loading and Previewing Data**

* **CSV Files:**  
  Use pd.read\_csv() to load tabular data from CSV files into a DataFrame.  
  Example:

python

**import** pandas **as** pd

df = pd.read\_csv('BigMart\_data.csv')

* **Preview:**  
  Use df.head() to see the first few rows and understand the structure and columns.

**2. Understanding Data Structure**

* DataFrames are tables with rows and columns, similar to Excel.
* Columns in your BigMart data include:  
  Item\_Identifier, Item\_Weight, Item\_Fat\_Content, Item\_Visibility, Item\_Type, Item\_MRP, Outlet\_Identifier, Outlet\_Establishment\_Year, Outlet\_Size, Outlet\_Location\_Type, Outlet\_Type, Item\_Outlet\_Sales.

**3. Subsetting Data**

* **Selecting Columns:**

python

df['Item\_Weight'] *# Single column (as Series)*

df[['Item\_Weight', 'Item\_Type']] *# Multiple columns (as DataFrame)*

* **Selecting Rows:**
  + By index: df.iloc (first row)
  + By condition: df[df['Outlet\_Size'] == 'Medium']

**4. Renaming Columns**

* Rename columns for clarity or consistency:

python

df.rename(columns={'Item\_Weight': 'Weight\_kg'}, inplace=True)

**5. Grouping and Aggregation**

* Calculate summary statistics (like mean sales) for groups:

python

sales\_by\_outlet = df.groupby('Outlet\_Size')['Item\_Outlet\_Sales'].mean()

* Count unique values per group:

python

unique\_items = df.groupby('Outlet\_Identifier')['Item\_Type'].nunique()

**6. Sorting Data**

* Sort values for better visualization or analysis:

python

sales\_by\_outlet.sort\_values(inplace=True)

**7. Handling Categorical Data**

* Convert categorical columns to numeric using:
  + **One-hot encoding:** pd.get\_dummies(df['Outlet\_Size'])
  + **Label encoding:** df['Outlet\_Size'].astype('category').cat.codes

**8. Working with Dates and Times**

* Convert strings to datetime:

python

df['Date'] = pd.to\_datetime(df['Date'], format='%d-%m-%Y %H:%M:%S')

* Extract date/time components:

python

df['Year'] = df['Date'].dt.year

df['Hour'] = df['Date'].dt.hour

**9. Visualization**

* **Bar Plot:**

python

plt.bar(x, y)

* **Scatter Plot:**

python

plt.scatter(x, y)

* **Boxplot:**

python

sns.boxplot(x='Outlet\_Size', y='Item\_Outlet\_Sales', data=df)

**10. Reading JSON Files**

* For standard JSON:

python

**import** json

**with** open('file.json') **as** f:

data = json.load(f)

df = pd.json\_normalize(data)

* For JSON Lines:

python

df = pd.read\_json('file.json', lines=True)

**Summary:**  
This assignment covers essential pandas techniques for reading, exploring, subsetting, transforming, and visualizing data, with practical examples on real-world retail data. Mastery of these steps is foundational for data analysis and machine learning workflows in Python.