

# Questions —> DATA QUALITY

**Question-1**, we were asked to list the data quality issues found in the dataset along with examples:

## 1. Data Quality Issue 1: Duplicate Transactions

Explanation:

Duplicate transactions refer to multiple entries with the same Transaction ID, indicating that the same transaction has been recorded multiple times in the dataset. This can lead to inaccurate analyses and reports, as it may result in overcounting sales or other metrics associated with the transactions. It is crucial to identify and resolve duplicate transactions to ensure the integrity and accuracy of the data.

Example:

```
aryansharma@Aryans-MacBook-Air G-7(ProcDNA) % python3 data_qua.py
Data Quality Issues:
Duplicate Transactions:
  Transaction ID Customer ID Customer Name Order Date ... Product Name Dollar Sales Returns Quantity
2      131065    AA-10375   Allen Arnold   9/7/22 ...   Blue Denim      30      0      2
3      131065    AA-10375   Allen Arnold  12/11/22 ...    Chinos       120      0      3
4      169488    AA-10480   Andrew Allen  7/17/21 ...    Chinos       120      0      3
5      169488    AA-10480   Andrew Allen  7/17/21 ...    Chinos        40      0      1
6      100230    AA-10480   Andrew Allen  7/17/21 ...   Blue Denim     120      0      8

[5 rows x 11 columns]

aryansharma@Aryans-MacBook-Air G-7(ProcDNA) % python3 data_qua.py
Data Quality Issues:
Duplicate Transactions:
  Transaction ID Customer ID Customer Name Order Date ... Product Name Dollar Sales Returns Quantity
2      131065    AA-10375   Allen Arnold   9/7/22 ...   Blue Denim      30      0      2
3      131065    AA-10375   Allen Arnold  12/11/22 ...    Chinos       120      0      3
4      169488    AA-10480   Andrew Allen  7/17/21 ...    Chinos       120      0      3
5      169488    AA-10480   Andrew Allen  7/17/21 ...    Chinos        40      0      1
6      100230    AA-10480   Andrew Allen  7/17/21 ...   Blue Denim     120      0      8
...      ...      ...      ...      ...      ...      ...      ...      ...
3424    102288    XP-21865   Xylona Preis  8/26/21 ...    Sneakers      150      0      3
3426    167682    YS-21880   Yana Sorensen 8/18/22 ...    Sneakers      150      0      3
3427    167682    ZC-21910   Zuschuss Carroll 3/8/21 ...   Formal Shoes   270     12      3
3429    152471    ZC-21910   Zuschuss Carroll 11/6/22 ...    Sneakers       50      0      1
3430    152471    ZD-21925   Zuschuss Donatelli 4/3/21 ...   Running Shoes  280      2      4

[2037 rows x 11 columns]
```

In this example, we can see that transactions with Transaction ID 131065 and 169488 are duplicated, indicating that they appear multiple times in the dataset.

## 2. Data Quality Issue 2: Date Format Consistency

Explanation:

Date format consistency refers to ensuring that all date values in the dataset are in a standardized and consistent format.

Inconsistent date formats can lead to errors in date-related calculations and comparisons. It is essential to convert all date values to a uniform format to maintain data accuracy and consistency.

Example:

**Date Format Consistency Issue: time data '14/11/21' does not match format '%m/%d/%y' (match)**

In this example, the date '14/11/21' does not match the expected format '%m/%d/%y'. The inconsistency in date formats can cause issues when performing date-based operations.

### 3. Data Quality Issue 3: Missing Values

Explanation:

Missing values occur when certain data fields are empty or null in the dataset. These missing values can affect the results of analyses and calculations, leading to incomplete or inaccurate insights. It is important to identify and handle missing values appropriately, either by imputing them with appropriate values or excluding them from specific analyses.

Example:

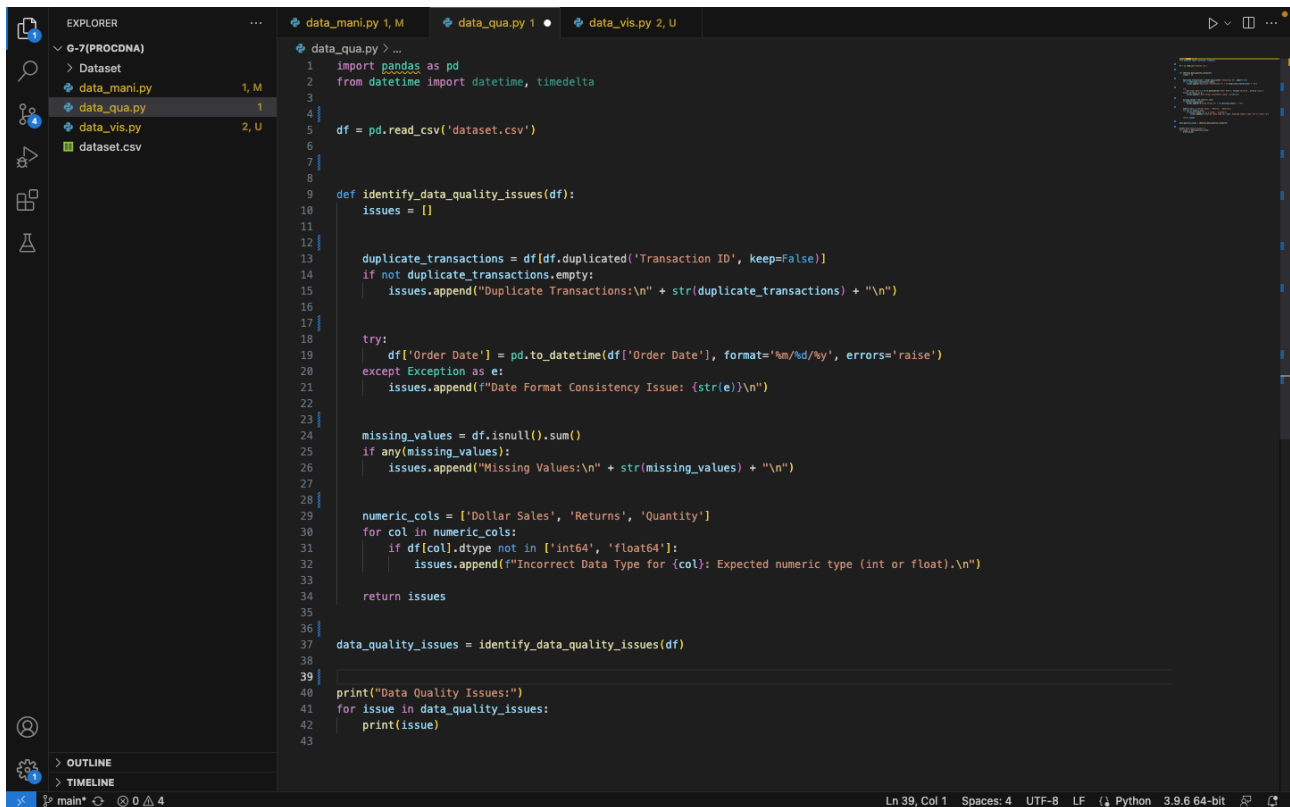
#### Missing Values

Transaction ID	0
Customer ID	0
Customer Name	0
Order Date	1
Sales Person Name	0
Product ID	0
Category	0
Product Name	0
Dollar Sales	0
Returns	0
Quantity	0
dtype	int64

In this example, there is one missing value in the 'Order Date' column. It is crucial to address this missing value to avoid potential errors in date-related analyses.

Addressing these data quality issues is crucial for maintaining data integrity, accuracy, and consistency in any data analysis or manipulation process. By resolving these issues, we can ensure that the results obtained from the dataset are reliable and meaningful.

**Question-2**, we were asked to Write a logic/algorithm to identify and report data quality issues:



The screenshot shows a VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'G-7(PROCDNA)' with a 'Dataset' folder containing 'data\_mani.py', 'data\_qua.py', 'data\_vis.py', and 'dataset.csv'. The code editor shows the 'data\_qua.py' file with the following Python code:

```
1 import pandas as pd
2 from datetime import datetime, timedelta
3
4 df = pd.read_csv('dataset.csv')
5
6
7
8
9 def identify_data_quality_issues(df):
10     issues = []
11
12
13     duplicate_transactions = df[df.duplicated('Transaction ID', keep=False)]
14     if not duplicate_transactions.empty:
15         issues.append("Duplicate Transactions:\n" + str(duplicate_transactions) + "\n")
16
17
18     try:
19         df['Order Date'] = pd.to_datetime(df['Order Date'], format='%m/%d/%y', errors='raise')
20     except Exception as e:
21         issues.append(f"Date Format Consistency Issue: {str(e)}\n")
22
23
24     missing_values = df.isnull().sum()
25     if any(missing_values):
26         issues.append("Missing Values:\n" + str(missing_values) + "\n")
27
28
29     numeric_cols = ['Dollar Sales', 'Returns', 'Quantity']
30     for col in numeric_cols:
31         if df[col].dtype not in ['int64', 'float64']:
32             issues.append(f"Incorrect Data Type for {col}: Expected numeric type (int or float).\n")
33
34     return issues
35
36
37 data_quality_issues = identify_data_quality_issues(df)
38
39
40 print("Data Quality Issues:")
41 for issue in data_quality_issues:
42     print(issue)
43
```

Data quality issues are already specified in “question-1”

### Algorithm:

1. Import the required libraries: pandas and datetime.
2. Read the dataset from 'dataset.csv' into a pandas DataFrame (df).
3. Define a function called 'identify\_data\_quality\_issues' that takes a DataFrame 'df' as input and returns a list of data quality issues.
  - a. Initialize an empty list called 'issues' to store the identified data quality issues.
  - b. Find duplicate transactions based on the 'Transaction ID' column using 'duplicated' method and store them in 'duplicate\_transactions' DataFrame.
  - c. If 'duplicate\_transactions' DataFrame is not empty, append the duplicate transactions information to the 'issues' list.
  - d. Convert the 'Order Date' column to pandas datetime format using 'pd.to\_datetime'. If any errors occur during the conversion, catch the exception and append the error message to the 'issues' list.

- e. Calculate the number of missing values for each column in the DataFrame using `'isnull().sum()'` and store them in the `'missing_values'` series.
  - f. If there are any missing values, append the information about missing values to the `'issues'` list.
  - g. Define a list called `'numeric_cols'` containing the names of columns expected to have numeric data (`'Dollar Sales'`, `'Returns'`, and `'Quantity'`).
  - h. Loop through each column in `'numeric_cols'` and check if its data type is not `'int64'` or `'float64'`. If the data type is not numeric, append the corresponding data type issue message to the `'issues'` list.
  - i. Return the `'issues'` list containing all the identified data quality issues.
4. Call the `'identify_data_quality_issues'` function passing the DataFrame `'df'` as an argument and store the returned list of data quality issues in `'data_quality_issues'`.
  5. Print the header message `"Data Quality Issues:"`.
  6. Loop through each issue in the `'data_quality_issues'` list and print each issue.

**Question-3**, we were asked to Identify critical quality issues impacting results in data manipulation questions:

Duplicate Transactions: If there are any rows with duplicate 'Transaction ID' values, the code will identify them and include the information about these duplicate transactions in the output, transactions with Transaction ID 131065 and 169488 are duplicated, indicating that they appear multiple times in the dataset. There were a total of 2037 rows having duplicate transactions.

Date Format Consistency Issue: The code attempts to convert the 'Order Date' column to pandas datetime format using the 'pd.to\_datetime' function with the format '%m/%d/%y' the date '14/11/21' does not match the expected format '%m/%d/%y'. The inconsistency in date formats can cause issues when performing date-based operations.

Missing Values: The code calculates the number of missing values for each column in the dataset using the 'isnull().sum()' method there is one missing value in the 'Order Date' column. It is crucial to address this missing value to avoid potential errors in date-related analyses