Name	**Bodhisatya Ghosh**
UID no.	**2021700026**

Experiment 5	
HONOUR PLEDGE	I hereby declare that the documentation, code and output attached with this lab experiment has been completed by me in accordance with the highest standards of honesty. I confirm that I have not plagiarized OR used unauthorized materials OR given or received illegitimate help for completing this experiment. I will uphold equity and honesty in the evaluation of my work and if found guilty of plagiarism or dishonesty, will bear the consequences as outlined in the 'integrity' section of the lab rubrics. I am doing so in order to maintain a community built around this code of honour. BGhosh Bodhisatya Ghosh
PROBLEM STATEMENT :	**Data Quality Assurance** **Answer the following question first:** **What are the 6 core dimensions of data quality? Elaborate on each with an example. Is there any other dimension that you can think of apart**

	from these 6 for measuring quality? **● Pick a time-series dataset from either of these websites based on batch majority:** **1) https://data.gov.in** **2) https://data.oecd.org** **3) Kaggle** **● Assess the quality of this dataset on all the dimensions of quality as identified by you.** **● Identify and handle inconsistent or erroneous data** **● Calculate data quality metrics for each data quality dimension for your dataset**
THEORY:	**1. Completeness** This dimension can cover a variety of attributes depending on the entity. For customer data, it shows the minimum information essential for a productive engagement. For example, if the customer address includes an optional landmark attribute, data can be considered complete even when the landmark information is missing. For products or services, data completeness can suggest vital attributes that help customers compare and choose. If a product description does not include any delivery estimate, it is not complete. Financial products often include historical performance details for customers to assess alignment with their requirements. Completeness measures if the data is sufficient to deliver meaningful inferences and decisions. 2. **Accuracy** Data accuracy is the level to which data represents the real-world scenario and confirms with a verifiable source. Accuracy of data ensures that the associated real-world entities can participate as planned. An accurate phone number of an employee guarantees that the employee is always reachable. Inaccurate birth details, on the other hand, can deprive the employee of

certain benefits.

Measuring data accuracy requires verification with authentic references such as birth records or with the actual entity. In some cases, testing can assure the accuracy of data. For example, you can verify customer bank details against a certificate from the bank, or by processing a transaction. Accuracy of data is highly impacted on how data is preserved through its entire journey, and successful data governance can promote this data quality dimension.

High data accuracy can power factually correct reporting and trusted business outcomes. Accuracy is very critical for highly regulated industries such as healthcare and finance.

3. **Consistency**

This data quality dimension represents if the same information stored and used at multiple instances matches. It is expressed as the percent of matched values across various records. Data consistency ensures that analytics correctly capture and leverage the value of data.

Consistency is difficult to assess and requires planned testing across multiple data sets. If one enterprise system uses a customer phone number with international code separately, and another system uses prefixed international code, these formatting inconsistencies can be resolved quickly. However, if the underlying information itself is inconsistent, resolving may require verification with another source. For example, if a patient record puts the date of birth as May 1st, and another record shows it as June 1st, you may first need to assess the accuracy of data from both sources. Data consistency is often associated with data accuracy, and any data set scoring high on both will be a high-quality data set.

4. **Validity**

This dimension signifies that the value attributes are available for aligning with the specific domain or requirement. For example, ZIP codes are valid

if they contain the correct characters for the region. In a calendar, months are valid if they match the standard global names. Using business rules is a systematic approach to assess the validity of data.

Any invalid data will affect the completeness of data. You can define rules to ignore or resolve the invalid data for ensuring completeness.

5. **Uniqueness**

This data quality dimension indicates if it is a single recorded instance in the data set used. Uniqueness is the most critical dimension for ensuring no duplication or overlaps. Data uniqueness is measured against all records within a data set or across data sets. A high uniqueness score assures minimized duplicates or overlaps, building trust in data and analysis.

Identifying overlaps can help in maintaining uniqueness, while data cleansing and deduplication can remediate the duplicated records. For example, unique customer profiles go a long way in offensive and defensive strategies for customer engagement. Data uniqueness also improves data governance and speeds up compliance.

6. **Integrity**

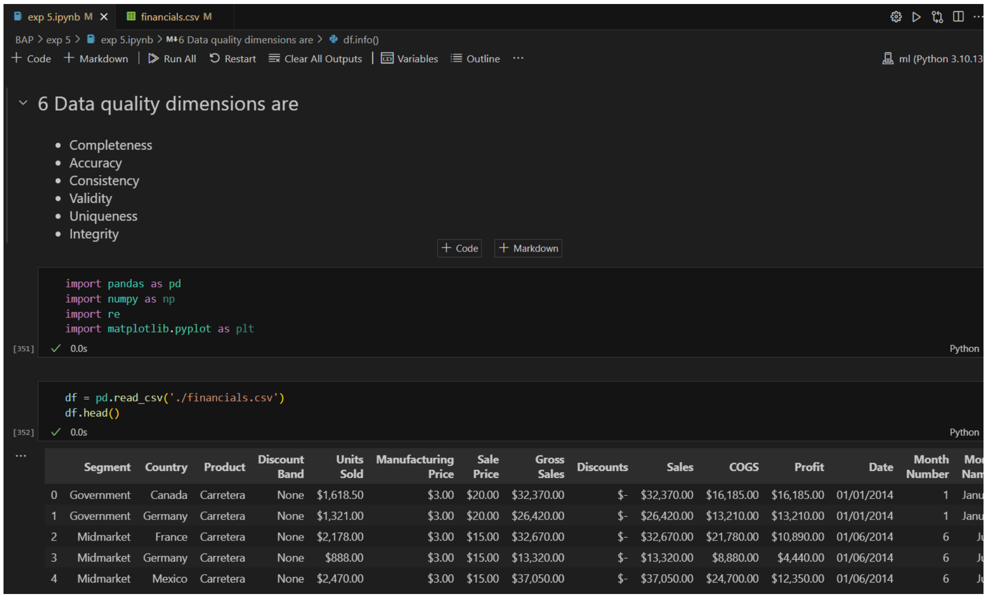
Data journey and transformation across systems can affect its attribute relationships. Integrity indicates that the attributes are maintained correctly, even as data gets stored and used in diverse systems. Data integrity ensures that all enterprise data can be traced and connected.

Data integrity affects relationships. For example, a customer profile includes the customer name and one or more customer addresses. In case one customer address loses its integrity at some stage in the data journey, the related customer profile can become incomplete and invalid.

While you regularly come across these six data quality dimensions and how they serve your company's needs, many more dimensions are available to

	represent distinctive attributes of data. Based on the context, you can also consider data conformity to standards (do data values comply with the specified formats?) for determining data quality. Data quality is multi-dimensional and closely linked with data intelligence, representing how your organization understands and uses data. PTO

RESULT:



6 Data quality dimensions are

- Completeness
- Accuracy
- Consistency
- Validity
- Uniqueness
- Integrity

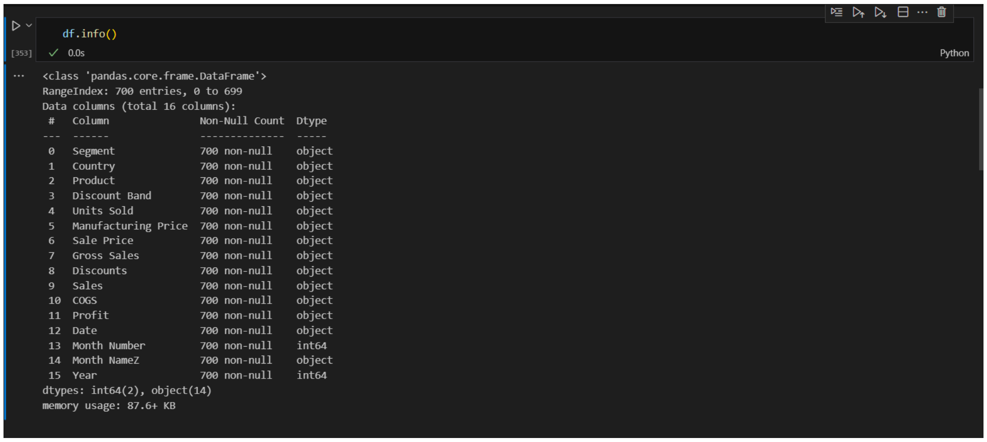
```
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
```

[351] ✓ 0.0s Python

```
df = pd.read_csv('./financials.csv')
df.head()
```

[352] ✓ 0.0s Python

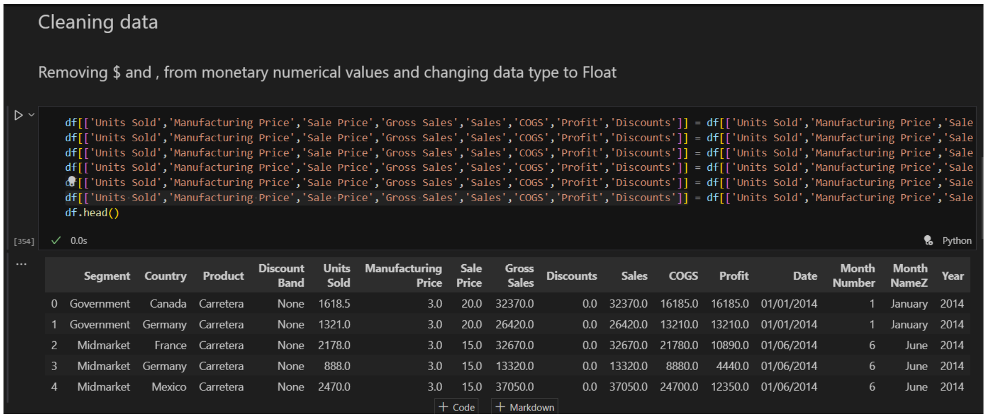
	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discounts	Sales	COGS	Profit	Date	Month Number	Mo Nam
0	Government	Canada	Carretera	None	$1,618.50	$3.00	$20.00	$32,370.00	$-	$32,370.00	$16,185.00	$16,185.00	01/01/2014	1	Janu
1	Government	Germany	Carretera	None	$1,321.00	$3.00	$20.00	$26,420.00	$-	$26,420.00	$13,210.00	$13,210.00	01/01/2014	1	Janu
2	Midmarket	France	Carretera	None	$2,178.00	$3.00	$15.00	$32,670.00	$-	$32,670.00	$21,780.00	$10,890.00	01/06/2014	6	Ju
3	Midmarket	Germany	Carretera	None	$888.00	$3.00	$15.00	$13,320.00	$-	$13,320.00	$8,880.00	$4,440.00	01/06/2014	6	Ju
4	Midmarket	Mexico	Carretera	None	$2,470.00	$3.00	$15.00	$37,050.00	$-	$37,050.00	$24,700.00	$12,350.00	01/06/2014	6	Ju



```
df.info()
```

[353] ✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   Segment              700 non-null    object
 1   Country              700 non-null    object
 2   Product              700 non-null    object
 3   Discount Band        700 non-null    object
 4   Units Sold           700 non-null    object
 5   Manufacturing Price  700 non-null    object
 6   Sale Price           700 non-null    object
 7   Gross Sales          700 non-null    object
 8   Discounts            700 non-null    object
 9   Sales                700 non-null    object
 10  COGS                 700 non-null    object
 11  Profit               700 non-null    object
 12  Date                 700 non-null    object
 13  Month Number         700 non-null    int64 
 14  Month NameZ          700 non-null    object
 15  Year                 700 non-null    int64 
dtypes: int64(2), object(14)
memory usage: 87.6+ KB
```



Cleaning data

Removing $ and , from monetary numerical values and changing data type to Float

```
df[['Units Sold','Manufacturing Price','Sale Price','Gross Sales','Sales','COGS','Profit','Discounts']] = df[['Units Sold','Manufacturing Price','Sale
df[['Units Sold','Manufacturing Price','Sale Price','Gross Sales','Sales','COGS','Profit','Discounts']] = df[['Units Sold','Manufacturing Price','Sale
df[['Units Sold','Manufacturing Price','Sale Price','Gross Sales','Sales','COGS','Profit','Discounts']] = df[['Units Sold','Manufacturing Price','Sale
df[['Units Sold','Manufacturing Price','Sale Price','Gross Sales','Sales','COGS','Profit','Discounts']] = df[['Units Sold','Manufacturing Price','Sale
df[['Units Sold','Manufacturing Price','Sale Price','Gross Sales','Sales','COGS','Profit','Discounts']] = df[['Units Sold','Manufacturing Price','Sale
df[['Units Sold','Manufacturing Price','Sale Price','Gross Sales','Sales','COGS','Profit','Discounts']] = df[['Units Sold','Manufacturing Price','Sale
df.head()
```

[354] ✓ 0.0s Python

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discounts	Sales	COGS	Profit	Date	Month Number	Month NameZ	Year
0	Government	Canada	Carretera	None	1618.5	3.0	20.0	32370.0	0.0	32370.0	16185.0	16185.0	01/01/2014	1	January	2014
1	Government	Germany	Carretera	None	1321.0	3.0	20.0	26420.0	0.0	26420.0	13210.0	13210.0	01/01/2014	1	January	2014
2	Midmarket	France	Carretera	None	2178.0	3.0	15.0	32670.0	0.0	32670.0	21780.0	10890.0	01/06/2014	6	June	2014
3	Midmarket	Germany	Carretera	None	888.0	3.0	15.0	13320.0	0.0	13320.0	8880.0	4440.0	01/06/2014	6	June	2014
4	Midmarket	Mexico	Carretera	None	2470.0	3.0	15.0	37050.0	0.0	37050.0	24700.0	12350.0	01/06/2014	6	June	2014

Replacing discount band with appropriate ordinal values

```python
print(df['Discount Band'].unique())
df['Discount Band'].replace({' None ':0,' Low ':1,' Medium ':2,' High ':3},inplace=True)
df.head()
```

[355] ✓ 0.0s Python

```
[' None ' ' Low ' ' Medium ' ' High ']
```

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discounts	Sales	COGS	Profit	Date	Month Number	Month NameZ	Year
0	Government	Canada	Carretera	0	1618.5	3.0	20.0	32370.0	0.0	32370.0	16185.0	16185.0	01/01/2014	1	January	2014
1	Government	Germany	Carretera	0	1321.0	3.0	20.0	26420.0	0.0	26420.0	13210.0	13210.0	01/01/2014	1	January	2014
2	Midmarket	France	Carretera	0	2178.0	3.0	15.0	32670.0	0.0	32670.0	21780.0	10890.0	01/06/2014	6	June	2014
3	Midmarket	Germany	Carretera	0	888.0	3.0	15.0	13320.0	0.0	13320.0	8880.0	4440.0	01/06/2014	6	June	2014
4	Midmarket	Mexico	Carretera	0	2470.0	3.0	15.0	37050.0	0.0	37050.0	24700.0	12350.0	01/06/2014	6	June	2014

Dropping month name as month number already given

+ Code + Markdown

Add Code Cell

```python
df.drop(columns=['Month NameZ'],inplace=True)
df.head()
```

[356] ✓ 0.0s Python

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discounts	Sales	COGS	Profit	Date	Month Number	Year
0	Government	Canada	Carretera	0	1618.5	3.0	20.0	32370.0	0.0	32370.0	16185.0	16185.0	01/01/2014	1	2014
1	Government	Germany	Carretera	0	1321.0	3.0	20.0	26420.0	0.0	26420.0	13210.0	13210.0	01/01/2014	1	2014

Changing date column data type

```python
df['Date'] = pd.to_datetime(df['Date'])
df.head()
```

[357] ✓ 0.0s Python

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discounts	Sales	COGS	Profit	Date	Month Number	Year
0	Government	Canada	Carretera	0	1618.5	3.0	20.0	32370.0	0.0	32370.0	16185.0	16185.0	2014-01-01	1	2014
1	Government	Germany	Carretera	0	1321.0	3.0	20.0	26420.0	0.0	26420.0	13210.0	13210.0	2014-01-01	1	2014
2	Midmarket	France	Carretera	0	2178.0	3.0	15.0	32670.0	0.0	32670.0	21780.0	10890.0	2014-01-06	6	2014
3	Midmarket	Germany	Carretera	0	888.0	3.0	15.0	13320.0	0.0	13320.0	8880.0	4440.0	2014-01-06	6	2014
4	Midmarket	Mexico	Carretera	0	2470.0	3.0	15.0	37050.0	0.0	37050.0	24700.0	12350.0	2014-01-06	6	2014

Data quality assurance

1. Completeness

It can be said that the data is complete as all needed parameters for product sales such as 'Discount Band', 'Units Sold','Manufacturing Price', 'Sale Price', 'Gross Sales', 'Discounts','Sales', 'COGS', 'Profit' and 'Date' are given

2. Accuracy

There is currently no way for verifying accuracy of this dataset

3. Consistency

There is currently no way for verifying consistency of this dataset

4. Validity

Considering profit as a validity parameter

4. Validity

Considering profit as a validity parameter

```
inconsistent = df[round(df['Profit'],2) != abs(round(df['Sales'] - df['COGS'], 2))]
print('{} entries are inconsistent according to profit-sales tally'.format(len(inconsistent)))
inconsistent
```

[358] ✓ 0.0s Python

```
0 entries are inconsistent according to profit-sales tally
```

Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discounts	Sales	COGS	Profit	Date	Month Number	Year

5. Uniqueness

```
df[df.duplicated()]
```

[359] ✓ 0.0s Python

Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discounts	Sales	COGS	Profit	Date	Month Number	Year

Dataset only contains unique datapoints

6. Integrity

There is currently no way for verifying integrity of this dataset

REFERENCES:

[The 6 Data Quality Dimensions with Examples | Collibra]

CONCLUSION: