Data Cleaning Project:

Introduction:

In the financial industry, the accuracy and reliability of data are critical. Financial statements, including the income statement, balance sheet, and cash flow statement, often come from multiple sources and may contain inconsistencies, missing values, or incorrect formats. This project aims to clean and prepare a raw financial dataset for analysis and reporting, ensuring data integrity for better financial decision-making.

Data Cleaning Process:

Data Import & Initial Exploration:

- **Loaded datasets using pandas**
-
- **Reviewed data types, structure, and summary**
- **Statistics**

Handling Missing Data:

- **Identified columns with missing values**
-
- **Filled missing values using appropriate techniques:**
-
    - **Forward-fill for time-series data**
    -
    - **Mean/median imputation for numerical data**
    -
    - **Dropped irrelevant or highly incomplete columns**

Cleaned Dataset Summary:

- **All columns correctly typed (numerical, categorical, dates)**
-
- **No missing or invalid values in key financial metrics**
-
- **Ready for use in dashboards, modeling, or financial forecasting**

Why is Data Cleaning Important?

Data cleaning is critically important in financial statements because these documents are used to make major business decisions, comply with regulations, and evaluate the financial health of a company. Here's why data cleaning matters:

Common Data Cleaning Tasks in Financial Statements:

- **Removing duplicate entries**
-
- **Correcting misclassified transactions**
-
- **Filling missing values or dates**
-
- **Ensuring format consistency (e.g., date, currency)**
-
- **Matching ledger accounts accurately**

In short, clean financial data = trustworthy financial reports, which is the foundation of all strategic financial activities.

How to Clean Data?

Cleaning data in a financial statement involves systematic steps to ensure all financial records are accurate, complete, and consistent before analysis or reporting. Here's a step-by-step guide on how to clean financial statement data:

1. **Import and Review the Raw Data**
2. **Remove Duplicates**
3. **Handle Missing Values**
4. **Standardize Formats**
5. **Validate Transaction Entries**
6. **Correct Misclassified Transactions**
7. **Reconcile with Source Documents**
8. **Check for Outliers or Anomalies**
9. **Audit Trail and Documentation**
10. **Automate Where Possible**

Financial Statement System Content:

Financial Analysis Project aims to analyze financial data using python's *panda's library*. The dataset contains various financial Records including Transaction ID, Date, Account Number, Amount, Category, Payment Mode Status Reference No Remarks Branch ***for a business or individual over a year.***

Read Excel File:

A simple way to store big data sets is to use Excel files (common separated files). Excel files contains plain text and is a well-known format that be read by everyone including pandas.

```
import pandas as pd
df=pd.read_excel('/content/unordered_finance_data_500.xlsx')
print(df.to_string())
```

VIEWING THE DATA: Show first 5 rows:

```
df.head()
```

Transaction ID	Date	Account Number	Amount	Category	Payment Mode	Status	Reference	Remarks	Branch	
0	435674	2023-04-01	49816537	-2058.32	Salary	CASH	Failed	716410	Delayed	Mumbai
1	421879	2023-05-24	93031557	724.28	Food	Credit Card	Failed	293392	Delayed	Tokyo
2	209751	2023-05-23	82507733	-1421.99	Misc	Credit Card	Pending	Original	NaN	New York
3	221958	2023-06-29	92649048	-1871.45	Salary	Debit Card	Failed	523602	Urgent	London
4	335362	2023-02-17	32564965	1490.75	Food	Bank Transfer	Pending	832081	NaN	New York

Show Last 5 Rows:

```
df.tail()
```

	Transaction ID	Date	Account Number	Amount	Category	Payment Mode	Status	Reference	Remarks	Branch
495	921654	2023-08-12	20016389	4087.01	Food	Credit Card	Completed	523966	Verified	Tokyo
496	693128	2023-01-19	90908854	-3145.41	Misc	Bank Transfer	Completed	958286	Verified	London
497	573254	2023-09-04	49138695	-4548.99	Salary	Credit Card	Completed	389954	Urgent	London
498	284779	2023-10-02	42359860	-380.57	Food	Bank Transfer	Completed	221668	Delayed	Mumbai
499	477812	2023-07-14	81531239	4786.31	Misc	Cash	Completed	864265	Delayed	Tokyo

Show how many rows and columns:

```
df.shape
```

```
(500, 10)
```

```
df.columns
```

```
Index(['Transaction ID', 'Date', 'Account Number', 'Amount', 'Category',
       'Payment Mode', 'Status', 'Reference', 'Remarks', 'Branch'],
      dtype='object')
```

Show no of count colunm,non null, data types:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Transaction ID  500 non-null    int64
 1   Date            500 non-null    datetime64[ns]
 2   Account Number  500 non-null    int64
 3   Amount          500 non-null    float64
 4   Category        500 non-null    object
 5   Payment Mode    500 non-null    object
 6   Status          500 non-null    object
 7   Reference       500 non-null    object
 8   Remarks         485 non-null    object
 9   Branch          500 non-null    object
dtypes: datetime64[ns](1), float64(1), int64(2), object(6)
memory usage: 39.2+ KB
```

Show only Branch:

```
df['Branch']
```

Show only unique Branch:

```
df['Branch'].unique()
```

Show no of count Branch:

```
df['Branch'].value_counts()
```

Show and count no of times null values:

```
df.isnull()
```

```
df.notnull().sum()
```

Remove null values:

```
df.dropna()
```

Drop the null vaues in original table:

```
df.dropna(inplace=True)
```

Fillna():This way you do not have to delete entire rows just because of some empty cells.

The fillna () method allows us to replace empty cells with value.

```
df.fillna(0)
```

```
df.fillna(0,inplace=True)
print(df)
```

Replace using Mean,Median (or) Mode:

```
x=df['Remarks']
print(x)
df['Remarks'].fillna(x,inplace=True)
print(df)
```

```
x=df['Payment Mode'].mode()
print(x)
df['Payment Mode'].fillna(x,inplace=True)
print(df)
```

Remove Duplicates:

```
print(df.duplicated())
```

```
df.drop_duplicates()
```

Rename:

```
df.rename(columns={'payment mode':'mode'})
```

Delete A Column:

```
df.drop('Date',axis=1)
```

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

```
df.head(10)
```

Modify Column :

```
df['Account Number']=df['Account Number']+111
```

Conclusion:

Data cleaning is a vital step in any financial analysis workflow. By resolving data integrity issues in the raw financial statements, we ensured that the final dataset is trustworthy and analysis-ready. The cleaned data can now be used to generate financial ratios, perform trend analysis, and support strategic business decisions with confidence.