Preprocess the cafe sale

The title for Cafe Sale dataset contains 10,000 transaction records from a café, detailing items sold, quantities, pricing, payment methods, locations, and transaction dates. The data includes inconsistencies such as missing values, errors (e.g., "ERROR" in numerical fields), and unspecified categories ("UNKNOWN"). This dataset is useful for practicing data cleaning, data wrangling, and sales analysis to derive insights into café operations and customer purchasing

1.Import libraries and datasets

```
import pandas as pd
df=pd.read_csv('/content/clean_cafe_sales_1200_rows.csv')
print(df)
```

```
    Transaction ID      Item  Quantity  Price Per Unit  Total Spent  \
0      TXN_1961373    Coffee       2.0             2.0          4.0
1      TXN_4977031      Cake       4.0             3.0         12.0
2      TXN_7034554     Salad       2.0             5.0         10.0
3      TXN_3160411    Coffee       2.0             2.0          4.0
4      TXN_2602893  Smoothie       5.0             4.0         20.0
..             ...       ...       ...             ...          ...
905    TXN_6928137      Cake       3.0             3.0          9.0
906    TXN_3003809    Coffee       4.0             2.0          8.0
907    TXN_5913013    Cookie       2.0             1.0          2.0
908    TXN_7433269       Tea       1.0             1.5          1.5
909    TXN_3103972      Cake       1.0             3.0          3.0

     Payment Method  Location Transaction Date
0       Credit Card  Takeaway       2023-09-08
1              Cash  In-Store       2023-05-16
2           Unknown   Unknown       2023-04-27
3    Digital Wallet  In-Store       2023-06-11
4       Credit Card   Unknown       2023-03-31
..              ...       ...              ...
905  Digital Wallet  In-Store       2023-10-11
906            Cash   Unknown       2023-04-21
907     Credit Card  Takeaway       2023-11-29
908            Cash  In-Store       2023-05-18
909         Unknown   Unknown       2023-03-01

[910 rows x 8 columns]
```

2.Display the entire datasett

```
df=pd.read_csv('/content/clean_cafe_sales_1200_rows.csv')
print(df.to_string)
```

```
<bound method DataFrame.to_string of     Transaction ID      Item  Quantity  Price Per Unit  Total Spent  \
0      TXN_1961373    Coffee       2.0             2.0          4.0
1      TXN_4977031      Cake       4.0             3.0         12.0
2      TXN_7034554     Salad       2.0             5.0         10.0
3      TXN_3160411    Coffee       2.0             2.0          4.0
4      TXN_2602893  Smoothie       5.0             4.0         20.0
..             ...       ...       ...             ...          ...
905    TXN_6928137      Cake       3.0             3.0          9.0
906    TXN_3003809    Coffee       4.0             2.0          8.0
907    TXN_5913013    Cookie       2.0             1.0          2.0
908    TXN_7433269       Tea       1.0             1.5          1.5
909    TXN_3103972      Cake       1.0             3.0          3.0

     Payment Method  Location Transaction Date
0       Credit Card  Takeaway       2023-09-08
1              Cash  In-Store       2023-05-16
2           Unknown   Unknown       2023-04-27
3    Digital Wallet  In-Store       2023-06-11
4       Credit Card   Unknown       2023-03-31
..              ...       ...              ...
905  Digital Wallet  In-Store       2023-10-11
906            Cash   Unknown       2023-04-21
907     Credit Card  Takeaway       2023-11-29
908            Cash  In-Store       2023-05-18
909         Unknown   Unknown       2023-03-01

[910 rows x 8 columns]>
```

3.Display Top 10 rowst

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

	Transaction ID	Item	Quantity	Price Per Unit	Total Spent	Payment Method	Location	Transaction Date
0	TXN_1961373	Coffee	2.0	2.0	4.0	Credit Card	Takeaway	2023-09-08
1	TXN_4977031	Cake	4.0	3.0	12.0	Cash	In-Store	2023-05-16
2	TXN_7034554	Salad	2.0	5.0	10.0	Unknown	Unknown	2023-04-27
3	TXN_3160411	Coffee	2.0	2.0	4.0	Digital Wallet	In-Store	2023-06-11
4	TXN_2602893	Smoothie	5.0	4.0	20.0	Credit Card	Unknown	2023-03-31
5	TXN_4433211	NaN	3.0	3.0	9.0	Unknown	Takeaway	2023-10-06
6	TXN_6699534	Sandwich	4.0	4.0	16.0	Cash	Unknown	2023-10-28
7	TXN_2064365	Sandwich	5.0	4.0	20.0	Unknown	In-Store	2023-12-31
8	TXN_2548360	Salad	5.0	5.0	25.0	Cash	Takeaway	2023-11-07
9	TXN_7619095	Sandwich	2.0	4.0	8.0	Cash	In-Store	2023-05-03

4.Display bottom 10 rows

```
df.tail(10)
```

	Transaction ID	Item	Quantity	Price Per Unit	Total Spent	Payment Method	Location	Transaction Date
900	TXN_6806290	Salad	3.0	5.0	15.0	Unknown	Unknown	2023-05-02
901	TXN_5386752	Salad	4.0	5.0	20.0	Digital Wallet	In-Store	2023-12-07
902	TXN_1022523	Salad	4.0	5.0	20.0	Cash	In-Store	2023-11-25
903	TXN_3125997	Juice	4.0	3.0	12.0	Unknown	Unknown	2023-10-19
904	TXN_6202015	Cookie	2.0	1.0	2.0	Credit Card	Takeaway	2023-08-22
905	TXN_6928137	Cake	3.0	3.0	9.0	Digital Wallet	In-Store	2023-10-11
906	TXN_3003809	Coffee	4.0	2.0	8.0	Cash	Unknown	2023-04-21
907	TXN_5913013	Cookie	2.0	1.0	2.0	Credit Card	Takeaway	2023-11-29
908	TXN_7433269	Tea	1.0	1.5	1.5	Cash	In-Store	2023-05-18
909	TXN_3103972	Cake	1.0	3.0	3.0	Unknown	Unknown	2023-03-01

5.Show How many rows and columns in dataset

```
df.shape
```

```
(910, 8)
```

6.Display the column Names

```
df.columns
```

```
Index(['Transaction ID', 'Item', 'Quantity', 'Price Per Unit', 'Total Spent',
       'Payment Method', 'Location', 'Transaction Date'],
      dtype='object')
```

7.Show non null values

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 910 entries, 0 to 909
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Transaction ID    910 non-null    object 
 1   Item              835 non-null    object 
 2   Quantity          910 non-null    float64
 3   Price Per Unit    910 non-null    float64
 4   Total Spent       910 non-null    float64
 5   Payment Method    910 non-null    object 
 6   Location          910 non-null    object 
 7   Transaction Date  910 non-null    object 
dtypes: float64(3), object(5)
memory usage: 57.0+ KB
```

8.Show null values count

```
df.isnull().sum()
```

	0
Transaction ID	0
Item	75
Quantity	0
Price Per Unit	0
Total Spent	0
Payment Method	0
Location	0
Transaction Date	0

dtype: int64

9.Find duplicate dataset

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

```
0      False
1      False
2      False
3      False
4      False
       ...
905    False
906    False
907    False
908    False
909    False
Length: 910, dtype: bool
```

10.Find the mean,meadian,count values

```
df.describe()
```

	Quantity	Price Per Unit	Total Spent
count	910.000000	910.000000	910.000000
mean	3.025275	3.013187	9.132967
std	1.425996	1.288795	6.065866
min	1.000000	1.000000	1.000000
25%	2.000000	2.000000	4.000000
50%	3.000000	3.000000	8.000000
75%	4.000000	4.000000	12.000000
max	5.000000	5.000000	25.000000

11.Find location

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

```
array(['Takeaway', 'In-Store', 'Unknown'], dtype=object)
```

12.Rename the column name Total spent into Total amount

```
df.rename(columns={'Total Spent':"Total amount"}, inplace=True)
df
```

	Transaction ID	Item	Quantity	Price Per Unit	Total amount	Payment Method	Location	Transaction Date
0	TXN_1961373	Coffee	2.0	2.0	4.0	Credit Card	Takeaway	2023-09-08
1	TXN_4977031	Cake	4.0	3.0	12.0	Cash	In-Store	2023-05-16
2	TXN_7034554	Salad	2.0	5.0	10.0	Unknown	Unknown	2023-04-27
3	TXN_3160411	Coffee	2.0	2.0	4.0	Digital Wallet	In-Store	2023-06-11
4	TXN_2602893	Smoothie	5.0	4.0	20.0	Credit Card	Unknown	2023-03-31
...	...	...	...	...	...	...	...	...
905	TXN_6928137	Cake	3.0	3.0	9.0	Digital Wallet	In-Store	2023-10-11
906	TXN_3003809	Coffee	4.0	2.0	8.0	Cash	Unknown	2023-04-21
907	TXN_5913013	Cookie	2.0	1.0	2.0	Credit Card	Takeaway	2023-11-29
908	TXN_7433269	Tea	1.0	1.5	1.5	Cash	In-Store	2023-05-18
909	TXN_3103972	Cake	1.0	3.0	3.0	Unknown	Unknown	2023-03-01

910 rows × 8 columns

13.Display the Transaction ID in ascending order

```
df.sort_values('Transaction ID')
```

	Transaction ID	Item	Quantity	Price Per Unit	Total amount	Payment Method	Location	Transaction Date
315	TXN_1002457	Cookie	5.0	1.0	5.0	Digital Wallet	Takeaway	2023-09-29
575	TXN_1018880	Smoothie	5.0	4.0	20.0	Digital Wallet	Unknown	2023-07-15
902	TXN_1022523	Salad	4.0	5.0	20.0	Cash	In-Store	2023-11-25
636	TXN_1026827	Cookie	4.0	1.0	4.0	Digital Wallet	In-Store	2023-01-02
266	TXN_1040764	Coffee	3.0	2.0	6.0	Cash	Takeaway	2023-07-27
...	...	...	...	...	...	...	...	...
343	TXN_9940220	Juice	2.0	3.0	6.0	Unknown	Takeaway	2023-03-05
422	TXN_9956154	Cake	4.0	3.0	12.0	Unknown	In-Store	2023-05-11
671	TXN_9959813	Sandwich	4.0	4.0	16.0	Digital Wallet	Unknown	2023-09-05
562	TXN_9960577	Sandwich	1.0	4.0	4.0	Cash	Unknown	2023-06-24
76	TXN_9999113	Juice	4.0	3.0	12.0	Cash	Takeaway	2023-05-27

910 rows × 8 columns

14.Fill missing numeric columns with appropriate values

```
df['Quantity'] = pd.to_numeric(df['Quantity'], errors='coerce').fillna(1)
df['Price Per Unit'] = pd.to_numeric(df['Price Per Unit'], errors='coerce').fillna(0)
df['Total amount'] = pd.to_numeric(df['Total amount'], errors='coerce')
df['Total amount'] = df['Total amount'].fillna(df['Quantity'] * df['Price Per Unit'])
print(df)
```

```
    Transaction ID      Item  Quantity  Price Per Unit  Total amount  \
0      TXN_1961373    Coffee       2.0             2.0           4.0
1      TXN_4977031      Cake       4.0             3.0          12.0
2      TXN_7034554     Salad       2.0             5.0          10.0
3      TXN_3160411    Coffee       2.0             2.0           4.0
4      TXN_2602893  Smoothie       5.0             4.0          20.0
..             ...       ...       ...             ...           ...
905    TXN_6928137      Cake       3.0             3.0           9.0
906    TXN_3003809    Coffee       4.0             2.0           8.0
907    TXN_5913013    Cookie       2.0             1.0           2.0
908    TXN_7433269       Tea       1.0             1.5           1.5
909    TXN_3103972      Cake       1.0             3.0           3.0

     Payment Method  Location Transaction Date
0       Credit Card  Takeaway       2023-09-08
1              Cash  In-Store       2023-05-16
2           Unknown   Unknown       2023-04-27
3    Digital Wallet  In-Store       2023-06-11
4       Credit Card   Unknown       2023-03-31
..              ...       ...              ...
905  Digital Wallet  In-Store       2023-10-11
906            Cash   Unknown       2023-04-21
907     Credit Card  Takeaway       2023-11-29
```

```
908            Cash  In-Store       2023-05-18
909         Unknown   Unknown       2023-03-01

[910 rows x 8 columns]
```

15.Fill empty cells using coloumn and replace the value:unkown in none

```
df['Item'] = df['Item'].fillna('Unknown')
df['Payment Method'] = df['Payment Method'].replace({'UNKNOWN': None}).fillna('Unknown')
df['Location'] = df['Location'].replace({'UNKNOWN': None}).fillna('Unknown')
df['Transaction Date'] = df['Transaction Date'].fillna('Unknown')
print(df.to_string)
```

```
df.columns=df.columns.str.replace(' ','_')
print(df)
```

```
df.to_csv('cleaned_cafe_sales.csv',index=False)
```

```
df.isnull().sum()
```

Conclusion

The cleaned café sales dataset provides clear, organized insights into the business's daily operations. It records detailed information on items sold, quantities, unit prices, total spending, payment methods, sales locations, and transaction dates. This data can help identify popular products, monitor sales trends, analyze customer payment preferences, and spot areas for operational improvement. Overall, it serves as a reliable foundation for further sales analysis, forecasting, and strategic decision-making to enhance the café's performance and customer satisfaction.