Import Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the input files

```
tran = pd.read_csv('TRANSACTION_TAKEHOME.csv')
tran.head()
```

	RECEIPT_ID	PURCHASE_DATE	SCAN_DATE	STORE_NAME	USER_ID	BARCODE	FINAL_QUANTITY	FINAL_SALE
0	0000d256-4041-4a3e-adc4-5623fb6e0c99	2024-08-21	2024-08-21 14:19:06.539 Z	WALMART	63b73a7f3d310dceeabd4758	1.530001e+10	1.00	
1	0001455d-7a92-4a7b-a1d2-c747af1c8fd3	2024-07-20	2024-07-20 09:50:24.206 Z	ALDI	62c08877baa38d1a1f6c211a	NaN	zero	1.49
2	00017e0a-7851-42fb-bfab-	2024-08-18	2024-08-19	WALMART	60842f207ac8b7729e472020	7.874223e+10	1.00	

Next steps: Generate code with tran | View recommended plots | New interactive sheet

```
tran.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   RECEIPT_ID      50000 non-null  object 
 1   PURCHASE_DATE   50000 non-null  object 
 2   SCAN_DATE       50000 non-null  object 
 3   STORE_NAME      50000 non-null  object 
 4   USER_ID         50000 non-null  object 
 5   BARCODE         44238 non-null  float64
 6   FINAL_QUANTITY  50000 non-null  object 
 7   FINAL_SALE      50000 non-null  object 
dtypes: float64(1), object(7)
memory usage: 3.1+ MB
```

Converting the data types

```
tran['BARCODE'] = tran['BARCODE'].fillna(0).astype(int)
tran.head()
```

	RECEIPT_ID	PURCHASE_DATE	SCAN_DATE	STORE_NAME	USER_ID	BARCODE	FINAL_QUANTITY	FINAL_SALE
0	0000d256-4041-4a3e-adc4-5623fb6e0c99	2024-08-21	2024-08-21 14:19:06.539 Z	WALMART	63b73a7f3d310dceeabd4758	15300014978	1.00	
1	0001455d-7a92-4a7b-a1d2-c747af1c8fd3	2024-07-20	2024-07-20 09:50:24.206 Z	ALDI	62c08877baa38d1a1f6c211a	0	zero	1.49
2	00017e0a-7851-42fb-bfab-	2024-08-18	2024-08-19	WALMART	60842f207ac8b7729e472020	78742229751	1.00	

Next steps: Generate code with tran | View recommended plots | New interactive sheet

Converting the data type for BARCODE column into integer as it is of type float.

Missing data count

```
tran.isnull().sum()
```

	0
RECEIPT_ID	0
PURCHASE_DATE	0
SCAN_DATE	0
STORE_NAME	0
USER_ID	0
BARCODE	0
FINAL_QUANTITY	0
FINAL_SALE	0

dtype: int64

Although the results shows there are no values, upon manual inspection of data we observe several empty values in FINAL_SALE, FINAL_QUANTITY columns.

Checking for duplicates

```
tran.duplicated().sum()
```

```
171
```

Around 171 Duplicate rows were observed in TRANSACTION dataset

Unique values in columns

```
tran['RECEIPT_ID'].is_unique
```

```
False
```

This indicates that the Receipt_ID column doesn't contain unique values.

```
tran['FINAL_QUANTITY'].unique()
```

```
array(['1.00', 'zero', '2.00', '3.00', '4.00', '4.55', '2.83', '2.34',
       '0.46', '7.00', '18.00', '12.00', '5.00', '2.17', '0.23', '8.00',
       '1.35', '0.09', '2.58', '1.47', '16.00', '0.62', '1.24', '1.40',
       '0.51', '0.53', '1.69', '6.00', '2.39', '2.60', '10.00', '0.86',
       '1.54', '1.88', '2.93', '1.28', '0.65', '2.89', '1.44', '2.75',
       '1.81', '276.00', '0.87', '2.10', '3.33', '2.54', '2.20', '1.93',
       '1.34', '1.13', '2.19', '0.83', '2.61', '0.28', '1.50', '0.97',
       '0.24', '1.18', '6.22', '1.22', '1.23', '2.57', '1.07', '2.11',
       '0.48', '9.00', '3.11', '1.08', '5.53', '1.89', '0.01', '2.18',
       '1.99', '0.04', '2.25', '1.37', '3.02', '0.35', '0.99', '1.80',
       '3.24', '0.94', '2.04', '3.69', '0.70', '2.52', '2.27'],
      dtype=object)
```

```
tran['FINAL_SALE'].unique()
```

```
array([' ', '1.49', '3.49', ..., '11.02', '20.17', '42.38'], dtype=object)
```

FINAL_SALE and FINAL_QUANTITY columns contain several null values

Investigating issues with data

```
tran.head(10)
```

	RECEIPT_ID	PURCHASE_DATE	SCAN_DATE	STORE_NAME	USER_ID	BARCODE	FINAL_QUANTITY	FINAL_SALE
0	0000d256-4041-4a3e-adc4-5623fb6e0c99	2024-08-21	2024-08-21 14:19:06.539 Z	WALMART	63b73a7f3d310dceeabd4758	15300014978	1.00	
1	0001455d-7a92-4a7b-a1d2-c747af1c8fd3	2024-07-20	2024-07-20 09:50:24.206 Z	ALDI	62c08877baa38d1a1f6c211a	0	zero	1.49
2	00017e0a-7851-42fb-bfab-0baa96e23586	2024-08-18	2024-08-19 15:38:56.813 Z	WALMART	60842f207ac8b7729e472020	78742229751	1.00	
3	000239aa-3478-453d-801e-66a82e39c8af	2024-06-18	2024-06-19 11:03:37.468 Z	FOOD LION	63fcd7cea4f8442c3386b589	783399746536	zero	3.49
4	00026b4c-dfe8-49dd-b026-4c2f0fd5c6a1	2024-07-04	2024-07-05 15:56:43.549 Z	RANDALLS	6193231ae9b3d75037b0f928	47900501183	1.00	

Next steps: Generate code with tran | View recommended plots | New interactive sheet

```
receipt_count = tran.groupby('RECEIPT_ID').size().reset_index(name='Duplicate_Count')
any_odd = (receipt_count['Duplicate_Count'] % 2 != 0).any()
if not any_odd:
    print("All even")
else:
    print("All odd")
```

```
All even
```

Each Receipt id entry is repeated twice with same values in all columns except FINAL_QUANTIY, FINAL_PRICE and as observed below one of the entries contain FINAL_SALE values as null.

```
tran[tran['RECEIPT_ID'] == '0000d256-4041-4a3e-adc4-5623fb6e0c99']
```

	RECEIPT_ID	PURCHASE_DATE	SCAN_DATE	STORE_NAME	USER_ID	BARCODE	FINAL_QUANTITY	FINAL_SALE
0	0000d256-4041-4a3e-adc4-5623fb6e0c99	2024-08-21	2024-08-21 14:19:06.539 Z	WALMART	63b73a7f3d310dceeabd4758	15300014978	1.00	

Below are the issues observed in the TRANSACTIONS dataset:

1) BARCODE column contains several null values and is of float datatype

2) Dataset contains 215 duplicate rows.

3) Dataset contains dupliate rows which only differ by FINAL_QUANTITY and FINAL_SALE values.

The below are the challenges observed with the dataset fields.

1) each receipt id contains two entries which only differ on quantity and sale values. Further calrification is to be given on which rows are to be used in the dataset for the which purpose.

2) SCAN_DATE and PURCHASE_DATE are not identical more calrification needed on the SCAN_DATE column definition.

3) This Dataset doesn't have a primary Key, hence this dataset can be further divided into 2 dataset with columns:

- **RECEIPT_ID, PURCHASE_DATE, SCAN_DATE, STORE_NAME, USER_ID**
- **RECEIPT_ID, BARCODE, FINAL_QUANTITY, FINAL_SALE**

All columns with no FINAL_SALE value can be ignored but this effects the FINAL_QUANTITY value.

All the above mentioned changes are to be applied before using this dataset for further steps in the process.