Clean Data

1. Read the data and check missing values
    a. In order to clean the data, the very first step involves loading the dataset into a Pandas DataFrame using the "**pd.read_csv** " function. Then, "**data.info()"** is used to get a concise summary of the DataFrame, particularly to identify columns with missing (null) values which, in this case is **Merchnum**, **Merch state**, and **Merch zip**.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 97852 entries, 0 to 97851
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Recnum             97852 non-null  int64
 1   Cardnum            97852 non-null  int64
 2   Date               97852 non-null  datetime64[ns]
 3   Merchnum           94455 non-null  object
 4   Merch description  97852 non-null  object
 5   Merch state        96649 non-null  object
 6   Merch zip          93149 non-null  float64
 7   Transtype          97852 non-null  object
 8   Amount             97852 non-null  float64
 9   Fraud              97852 non-null  int64
dtypes: datetime64[ns](1), float64(2), int64(3), object(4)
memory usage: 7.5+ MB
```

```
Recnum                  0
Cardnum                 0
Date                    0
Merchnum             3220
Merch description       0
Merch state          1028
Merch zip            4347
Transtype               0
Amount                  0
Fraud                   0
dtype: int64
```

2. Clean and impute **Merchnum**
    a. We have identified a total of *3,220* instances where the **Merchnum** attribute is not available. Our objective is to substitute these missing values with the best possible estimates.
        i. Initially, we employed the **Merch description** attribute to deduce the corresponding **Merchnum** value. This strategy enabled us to address *1,164* cases; however, we were left with *2,115* records still lacking a **Merchnum**.
        ii. Upon encountering records with the **Merch description** indicating ***'RETAIL CREDIT ADJUSTMENT'***, we categorized the **Merchnum** as ***'unknown'***. This method resolved another *694* cases, which reduced the number of records missing a **Merchnum** to *1,421*.
        iii. For the remaining *1,421* records without a **Merchnum**, we noted a diversity in the **Merch description** attribute, comprising *515* unique **merchant descriptions**. These descriptions likely correspond to various merchants, each with a small number of transactions. Consequently, we assigned a distinct and novel **Merchnum** to each unique merchant description. Following this process, all records were supplemented with a valid **Merchnum** value, ensuring that the dataset no longer contained any missing **Merchnum** data.

3. Clean and impute **Merch state**
    a. In the initial assessment of the dataset, we observed that the **Merch state** field was missing for *1,028* records. To address this, we first examined records where **Merch state** was null to understand the pattern of missing data.
    b. A notable finding was that transactions with the **Merch description** of ***'RETAIL DEBIT ADJUSTMENT' or 'RETAIL CREDIT ADJUSTMENT'*** often lacked a corresponding **Merch state**. For such cases, we decided to impute the state as ***'unknown'***.
    c. Next, we created mappings based on available data:
        i. A dictionary to map **Merch zip** codes to states (**zip_state**), allowing us to impute missing state values based on zip codes. With this approach, we succeeded in diminishing the missing **Merch state** values from 1*,028* to *954*, thus making considerable progress.
        ii. Further, we constructed two additional mappings, one (**merchnum_state**) relating **Merchnum** to **Merch state** and the other (**merchdes_state**) associating **Merch description** to **Merch state**. The application of these mappings resulted in the reduction of missing state values from *954* to *953* and then to *952*, respectively, after each mapping was applied. This method is not very effective obviously.
    d. For the remaining cases, we re-established a rule: any transaction marked as an adjustment (***'RETAIL CREDIT ADJUSTMENT' or 'RETAIL DEBIT ADJUSTMENT'***) in the **Merch description** would have its **Merch state** imputed as ***'unknown'***. This strategy further reduced the missing values to *297*.
    e. Upon inspection, it was evident that some of the **Merch state** entries contained non-U.S. state codes. In an effort to maintain dataset uniformity, we relabeled these as ***'foreign'***. The final act of our imputation process was to assign ***'unknown'*** to any residual nulls in the **Merch state** field.
    f. Following this process, all records were supplemented with a valid **Merch state** info, ensuring that the dataset no longer contained any missing **Merch state** data.
4. Clean and impute **Merch zip**
    a. In confronting the issue of missing **'Merch zip'** codes within our dataset, we noted *4,347* instances where this data was absent. The initial step involved creating associative mappings from **'Merchnum'** and **'Merch description**' to their corresponding zip codes, where available. This allowed us to deduce and fill in a large number of missing zip codes, which reduced the number of nulls in the '**Merch zip'** field to *2,625*.
    b. Moving forward, for records that had a valid **'Merch state'** but lacked a **'Merch zip**', we employed a strategic imputation using the most populous zip code within the given state. This data was compiled into the **mostPopZip** dictionary, sourced

    from an external reference. The application of this approach saw the number of missing zip codes drop to *1,216*.
    c. To address the final missing entries, we took the conservative step of assigning the value ***'unknown'*** to all remaining nulls in the **'Merch zip**' field. This action resolved the issue entirely, ensuring that every record in our dataset had a complete set of data for the **'Merch zip'** field, thereby enhancing the integrity and usability of our dataset for any further analysis.
    d. Following this process, all records were supplemented with a valid **'Merch zip** info, ensuring that the dataset no longer contained any missing **'Merch zip** data.
5. Double check null values
    a. After carefully exclusions, outlier treatment and imputation, we got a non-null dataset that enhancing the high quality and usability for any future using and analyzing.

```
1  data.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
Index: 97496 entries, 0 to 97851
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Recnum             97496 non-null  int64
 1   Cardnum            97496 non-null  int64
 2   Date               97496 non-null  datetime64[ns]
 3   Merchnum           97496 non-null  object
 4   Merch description  97496 non-null  object
 5   Merch state        97496 non-null  object
 6   Merch zip          97496 non-null  object
 7   Transtype          97496 non-null  object
 8   Amount             97496 non-null  float64
 9   Fraud              97496 non-null  int64
dtypes: datetime64[ns](1), float64(1), int64(3), object(5)
memory usage: 10.2+ MB
```