# Backtracking Reference Document STAM Phase 1

## Function: read\_excel

**Purpose**: Reads data from an Excel file and returns it as a pandas DataFrame.

### Steps or Methods Used:

* Import the pandas library as pd.
* Use the pd.read\_excel() function to read data from the specified file path into a DataFrame.
* Handle any exceptions and errors during file reading.
* Return the DataFrame containing the Excel data.

### Formula:

df = pd.read\_excel(file\_path)

## Function: read\_csv

Purpose: Reads data from a CSV file and returns it as a pandas DataFrame.

### Steps or Methods Used:

* Import the pandas library as pd.
* Use the pd.read\_csv() function to read data from the specified file path into a DataFrame.
* Handle any exceptions and errors during file reading.
* Return the DataFrame containing the CSV data.

### Formula:

df = pd.read\_csv(file\_path)

**Function: describe\_data**

Purpose: Provides a summary of the DataFrame, including basic information and a statistical description.

### Steps or Methods Used:

* Print basic DataFrame information using df.info().
* Print a statistical summary using df.describe() to display count, mean, standard deviation, min, max, and quartile values for each numeric column.

**Function: rename\_columns**

Purpose: Renames columns in a DataFrame.

### Steps or Methods Used:

* Prepare a renaming dictionary that maps old column names to new ones.
* Use df.rename(columns=new\_columns) to rename the columns.
* Handle any errors that may occur during renaming.

### Formula:

df = df.rename(columns=new\_columns)

## Function: change\_data\_types

Purpose: Changes the data types of specified columns in a DataFrame.

### Steps or Methods Used:

* Prepare a data type mapping dictionary that maps column names to target data types.
* Use df.astype(column\_types) to convert the specified columns to the desired types.
* Handle any errors that may occur if the conversion is not possible.

### Formula:

df = df.astype(column\_types)

## Function: bin\_data

Purpose: Bins data in a DataFrame column into specified categories.

### Steps or Methods Used:

* Define bin edges and labels for categorizing the data.
* Use pd.cut() to create the binned categories and add the resulting series as a new column to the DataFrame.
* Handle any issues during binning, such as incorrect bin specifications.

### Formula:

df[f'{column}\_binned'] = pd.cut(df[column], bins=bins, labels=labels)

## Function: replace\_values

Purpose: Replaces specific values in a DataFrame column with new values.

### Steps or Methods Used:

* Identify the target values to be replaced in the specified column.
* Use df[column\_name].replace(to\_replace, value) to replace the old values with the new ones.

### Formula:

df[column\_name] = df[column\_name].replace(to\_replace, value)

## Function: handle\_null\_values

Purpose: Handles null values in a DataFrame using different strategies like mean, median, mode, or dropping the null values.

### Steps or Methods Used:

* Select a strategy for handling null values (mean, median, mode, or drop).
* Apply the chosen method to handle null values.
* Handle any invalid strategies by printing an error message.

### Formula:

df = df.fillna(df.mean()) # or other methods depending on strategy

## Function: handle\_outliers

Purpose: Handles outliers in a DataFrame column using different statistical methods (IQR or Z-score).

### Steps or Methods Used:

* Choose a method (IQR or Z-score) for detecting outliers.
* Calculate outlier thresholds based on the chosen method.
* Filter out or handle rows containing outliers.
* Handle any invalid methods by printing an error message.

### Formula:

df = df[(np.abs(stats.zscore(df[column])) < 3)] # or other methods depending on strategy

## Function: sample\_data

Purpose: Samples data from a DataFrame based on a specified method.

### Steps or Methods Used:

* Choose a sampling method (random or stratified).
* Perform sampling using the chosen method.
* Handle any invalid methods by printing an error message.

### Formula:

df.sample(n=n, random\_state=42) # or other methods depending on strategy

## Function: sub\_setting

Purpose: Subsets the DataFrame based on a condition.

### Steps or Methods Used:

* Define the condition for filtering rows.
* Use df.query(condition) to filter the DataFrame based on the specified condition.

### Formula:

df.query(condition)

## Function: create\_new\_column

Purpose: Creates a new column in the DataFrame based on a calculation.

### Steps or Methods Used:

* Define the new column calculation logic.
* Assign the result of the calculation to the new column in the DataFrame.

### Formula:

df['new\_column'] = df['column1'] + df['column2'] # Example calculation