

Data cleaning is an essential step in data preprocessing. It involves identifying and correcting errors, inconsistencies, and inaccuracies in the data. Here's a step-by-step approach to clean the provided dataset:

### Step 1: Import necessary libraries and load the data

```
import pandas as pd
import numpy as np

# Load the data
data = pd.read_csv('data.csv')
```

### Step 2: Handle missing values

```
# Check for missing values
print(data.isnull().sum())

# Replace missing values in 'brokered_by' column with NaN
data['brokered_by'] = data['brokered_by'].replace(['-1', '-'], np.nan)

# Replace missing values in 'price' column with NaN
data['price'] = data['price'].replace(['-1', '-'], np.nan)

# Replace missing values in 'bed' column with NaN
data['bed'] = data['bed'].replace(['-1', 'nan'], np.nan)

# Replace missing values in 'bath' column with NaN
data['bath'] = data['bath'].replace(['-1', 'nan'], np.nan)

# Replace missing values in 'acre_lot' column with NaN
data['acre_lot'] = data['acre_lot'].replace(['-1', 'nan'], np.nan)

# Replace missing values in 'street' column with NaN
data['street'] = data['street'].replace(['-'], np.nan)

# Replace missing values in 'city' column with NaN
data['city'] = data['city'].replace(['-'], np.nan)

# Replace missing values in 'state' column with NaN
data['state'] = data['state'].replace(['-'], np.nan)

# Replace missing values in 'zip_code' column with NaN
data['zip_code'] = data['zip_code'].replace(['-1', '-'], np.nan)

# Replace missing values in 'house_size' column with NaN
data['house_size'] = data['house_size'].replace(['-1', '-'], np.nan)

# Replace missing values in 'prev_sold_date' column with NaN
data['prev_sold_date'] = data['prev_sold_date'].replace(['-'], np.nan)
```

### Step 3: Convert data types

```
# Convert 'price' column to numeric
data['price'] = pd.to_numeric(data['price'], errors='coerce')

# Convert 'bed' column to numeric
data['bed'] = pd.to_numeric(data['bed'], errors='coerce')

# Convert 'bath' column to numeric
data['bath'] = pd.to_numeric(data['bath'], errors='coerce')

# Convert 'acre_lot' column to numeric
```

```
data['acre_lot'] = pd.to_numeric(data['acre_lot'], errors='coerce')

# Convert 'house_size' column to numeric
data['house_size'] = pd.to_numeric(data['house_size'], errors='coerce')

# Convert 'zip_code' column to numeric
data['zip_code'] = pd.to_numeric(data['zip_code'], errors='coerce')
```

#### Step 4: Handle inconsistent data

```
# Replace 'two' with 2 in 'bed' column
data['bed'] = data['bed'].replace(['two', 'three', 'four', 'five'], [2, 3, 4, 5])

# Replace 'two' with 2 in 'bath' column
data['bath'] = data['bath'].replace(['two', 'three', 'four', 'five'], [2, 3, 4, 5])
```

#### Step 5: Remove duplicates

```
# Remove duplicates
data = data.drop_duplicates()
```

#### Step 6: Save the cleaned data

```
# Save the cleaned data
data.to_csv('cleaned_data.csv', index=False)
```

After performing these steps, the dataset should be cleaned and ready for further analysis. Note that the specific cleaning steps may vary depending on the nature of the data and the requirements of the analysis.