Certainly! Data wrangling involves cleaning and transforming raw data into a more usable format. Here are some common data wrangling steps that can be applied to this dataset:

1. **Identify and Handle Missing Values:**
   - Rows 0, 10, 22, 39, 44, 45, and 70 contain entirely missing values. These can be removed.
   - For other missing values, decide whether to fill them with a placeholder, mean, median, or mode, or to drop them based on the context.
2. **Data Type Conversion:**
   - Ensure that numerical columns (`price`, `acre_lot`, `zip_code`, `house_size`, `brokered_by`) are of the correct data type (e.g., float or integer).
   - Convert `prev_sold_date` to a datetime format for easier date manipulation.
3. **Split and Clean `bedrooms_bathrooms`:**
   - Split the `bedrooms_bathrooms` column into two separate columns: `bedrooms` and `bathrooms`.
   - Convert these new columns to integer data types.
4. **Standardize Address Format:**
   - Ensure that the `address` column is consistently formatted, though it appears to be well-structured already.
5. **Remove Duplicates:**
   - Check for and remove any duplicate rows.
6. **Outlier Detection:**
   - Identify any potential outliers in numerical columns, such as `price` or `acre_lot`, and decide on a strategy to handle them.
7. **Feature Engineering:**
   - Create new features if necessary, such as `price_per_sqft` by dividing `price` by `house_size`.
8. **Filter Data:**
   - Depending on the analysis goal, filter the dataset to include only relevant rows (e.g., only `for_sale` properties).

Here's a Python code snippet using pandas to perform some of these steps:

```python
import pandas as pd

# Load the dataset
data = pd.read_csv('your_dataset.csv')  # Replace with the actual file path

# Drop rows with all NaN values
data.dropna(how='all', inplace=True)

# Convert data types
data['prev_sold_date'] = pd.to_datetime(data['prev_sold_date'])
data['price'] = data['price'].astype(float)
data['acre_lot'] = data['acre_lot'].astype(float)
data['zip_code'] = data['zip_code'].astype(str)
data['house_size'] = data['house_size'].astype(float)
data['brokered_by'] = data['brokered_by'].astype(float)

# Split bedrooms and bathrooms
data[['bedrooms', 'bathrooms']] = data['bedrooms_bathrooms'].str.split(', ', expand=True)
data['bedrooms'] = data['bedrooms'].astype(int)
data['bathrooms'] = data['bathrooms'].astype(int)

# Remove duplicates
data.drop_duplicates(inplace=True)

# Feature engineering
data['price_per_sqft'] = data['price'] / data['house_size']

# Handle missing values (example: fill with median)
data.fillna(data.median(), inplace=True)

# Display the cleaned dataset
print(data.head())
```

This code provides a basic framework for data wrangling. Depending on the specific requirements and context of your analysis, you might need to adjust these steps.