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. **Data Cleaning**

- **Remove Duplicates**: Check for and remove any duplicate rows.
- **Handle Missing Values**: Identify any missing values and decide how to handle them (e.g., fill with a default value, remove the row, etc.).
- **Standardize Data Types**: Ensure that each column has a consistent data type (e.g., numeric columns should be integers or floats).

2. **Data Transformation**

- **Convert Currency**: If needed, convert the price to a consistent currency format (e.g., remove the $ sign and convert to a float).
- **Date Handling**: Combine `prev_sold_month`, `prev_sold_day`, and `prev_sold_year` into a single `prev_sold_date` column with a proper date format.
- **Normalize Text**: Standardize text data, such as converting all text to lowercase or title case for consistency.

3. **Feature Engineering**

- **Calculate Price per Square Foot**: Create a new column that calculates the price per square foot of the house.
- **Categorize Properties**: Create categories based on the number of bedrooms or bathrooms (e.g., "small", "medium", "large").
- **Geographical Features**: Extract city and state from the address for easier geographical analysis.

4. **Data Filtering**

- **Filter by Status**: Separate the dataset into properties that are "sold" and those that are "for_sale".
- **Price Range Filtering**: Filter properties within a certain price range if needed for specific analysis.

Example Code (using Python and Pandas)

Here's a basic example of how you might perform some of these steps using Python and Pandas:

```
import pandas as pd

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

# Remove duplicates
data = data.drop_duplicates()

# Handle missing values (example: fill with zero)
data = data.fillna(0)

# Convert price to float
data['price'] = data['price'].replace('[\$,]', '', regex=True).astype(float)

# Combine date columns into a single date column
data['prev_sold_date'] = pd.to_datetime(data[['prev_sold_year', 'prev_sold_month', 'prev_sold_day']])

# Calculate price per square foot
data['price_per_sqft'] = data['price'] / data['house_size']

# Extract city and state from address
data[['city', 'state']] = data['address'].str.extract(r',\s*([^,]+),\s*([A-Z]{2})')

# Filter by status
sold_properties = data[data['status'] == 'sold']
for_sale_properties = data[data['status'] == 'for_sale']
```

```
# Display the first few rows of the cleaned dataset
print(data.head())
```

This code provides a basic framework for data wrangling. Depending on your specific needs, you might want to add more transformations or filters.