Data standardization is the process of transforming data into a standard format to ensure consistency and accuracy. Here's a step-by-step guide on how to standardize the provided dataset:

Step 1: Handling Missing Values

After reviewing the dataset, it appears that there are no missing values. However, some values are represented in a non-standard format (e.g., 's' instead of 'sold', 'f' instead of 'for_sale', and numerical values represented as strings).

Step 2: Standardizing Status Column

The 'status' column contains values like 's', 'sold', 'f', and 'for_sale'. We'll standardize these values to 'sold' and 'for_sale'.

Step 3: Standardizing Bed and Bath Columns

The 'bed' and 'bath' columns contain numerical values represented as strings (e.g., 'three', 'two', 'five', 'four'). We'll standardize these values to integers.

Step 4: Standardizing Price Column

The 'price' column contains numerical values with a dollar sign. We'll remove the dollar sign and convert the values to integers.

Step 5: Standardizing Prev_Sold_Date Column

The 'prev_sold_date' column contains dates in different formats. We'll standardize these dates to a single format (e.g., YYYY-MM-DD).

Here's a Python code snippet using the pandas library to perform these standardization steps:

```
import pandas as pd

# Load the dataset
data = {
    # ... (your dataset here)
}

df = pd.DataFrame(data)

# Standardize status column
df['status'] = df['status'].apply(lambda x: 'sold' if x in ['s', 'sold'] else 'for_sale' if x in ['f', 'for_sa

# Standardize bed and bath columns
bed_map = {'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5}
df['bed'] = df['bed'].apply(lambda x: bed_map.get(str(x), x) if isinstance(x, int) else int(x))
df['bath'] = df['bath'].apply(lambda x: bed_map.get(str(x), x) if isinstance(x, int) else int(x))

# Standardize price column
df['price'] = df['price'].apply(lambda x: int(str(x).replace('$', '').replace(',', '')))

# Standardize prev_sold_date column
df['prev_sold_date'] = pd.to_datetime(df['prev_sold_date'], errors='coerce')

# Print the standardized dataset
print(df)
```

Note that this code assumes that the dataset is stored in a dictionary called `data`. You'll need to replace this with your actual dataset.

After running this code, the dataset should be standardized, with consistent formatting and no missing values.