I'll help you clean this dataset. Here are the main issues I notice and how we can clean them:

1. Standardize missing values (replace various forms like 'nan', '-1', '-', empty strings with proper NaN)
2. Convert numeric columns to proper data types
3. Standardize text values (like 'three' to '3', 'f' to 'for_sale')
4. Fix date formats
5. Clean up price values (remove '$' and standardize)
6. Standardize state abbreviations

Here's a Python script to clean the data:

```
import pandas as pd
import numpy as np

# Read the data
df = pd.read_csv('your_file.csv')

# Function to convert text numbers to numeric
def text_to_number(x):
    number_map = {
        'one': '1', 'two': '2', 'three': '3', 'four': '4', 'five': '5',
        'six': '6', 'seven': '7', 'eight': '8', 'nine': '9'
    }
    if isinstance(x, str):
        return number_map.get(x.lower(), x)
    return x

# Function to standardize status
def clean_status(x):
    if pd.isna(x):
        return np.nan
    x = str(x).lower()
    if x in ['f', 's']:
        return 'for_sale' if x == 'f' else 'sold'
    return x

# Clean missing values
df = df.replace(['nan', '-1', '-', '', 'Unknown'], np.nan)

# Clean status
df['status'] = df['status'].apply(clean_status)

# Clean numeric columns
df['bed'] = df['bed'].apply(text_to_number)
df['bath'] = df['bath'].apply(text_to_number)

# Convert numeric columns
numeric_columns = ['price', 'bed', 'bath', 'acre_lot', 'house_size']
for col in numeric_columns:
    df[col] = pd.to_numeric(df[col].str.replace('$', '').str.replace(',', ''), errors='coerce')

# Clean dates
df['prev_sold_date'] = pd.to_datetime(df['prev_sold_date'], format='mixed', errors='coerce')

# Clean state names (standardize to two-letter codes)
state_map = {
    'California': 'CA',
    'Texas': 'TX',
    'Florida': 'FL',
    'Massachusetts': 'MA',
    'South Carolina': 'SC',
```

```
    'Idaho': 'ID',
    'Maryland': 'MD',
    'Kentucky': 'KY',
    'Nevada': 'NV',
    'Indiana': 'IN',
    'Alabama': 'AL',
    'Virginia': 'VA',
    'Ohio': 'OH',
    'Illinois': 'IL',
    'Washington': 'WA',
    'Colorado': 'CO',
    'Montana': 'MT',
    'Tennessee': 'TN',
    'Missouri': 'MO',
    'Wisconsin': 'WI',
    'Kansas': 'KS',
    'Michigan': 'MI',
    'Oklahoma': 'OK',
    'Georgia': 'GA',
    'Minnesota': 'MN',
    'New Jersey': 'NJ',
    'Pennsylvania': 'PA',
    'Utah': 'UT'
}
df['state'] = df['state'].replace(state_map)

# Clean zip codes
df['zip_code'] = pd.to_numeric(df['zip_code'], errors='coerce')

# Drop duplicates if any
df = df.drop_duplicates()

# Reset index
df = df.reset_index(drop=True)
```

After cleaning, the data will have: - Consistent missing value representation (NaN) - Numeric values in proper format - Standardized text values - Consistent date format - Standardized state codes - Proper data types for each column

You might want to add additional cleaning steps depending on your specific needs, such as: 1. Handling outliers 2. Validating zip codes 3. Standardizing street addresses 4. Adding data validation rules 5. Handling specific business rules

Would you like me to elaborate on any of these aspects or provide additional cleaning steps?