

```

import pandas as pd

# Load the data into a Pandas DataFrame
data = """brokered_by,broker_first_name,broker_surname,status,price,currency,bed,bath,acre_lot,address,zip_code
94681,Lucas,Walker,sold,409900,$,2,1,2.12,"655 First St, Stockbridge, Massachusetts",1262,1015,11,29,2021
... (rest of your data) ...""" # Replace ... with your actual data
df = pd.read_csv(pd.compat.StringIO(data))

# 1. Data Cleaning:

# Remove currency column (it's all the same)
df.drop('currency', axis=1, inplace=True)

# Convert price to numeric, coercing errors to NaN (in case of non-numeric values)
df['price'] = pd.to_numeric(df['price'], errors='coerce')

# Convert bed, bath, acre_lot, house_size to numeric
numeric_cols = ['bed', 'bath', 'acre_lot', 'house_size']
df[numeric_cols] = df[numeric_cols].apply(pd.to_numeric, errors='coerce')

# Convert zip_code to string (to preserve leading zeros if needed)
df['zip_code'] = df['zip_code'].astype(str)

# Create a proper datetime column for prev_sold_date
df['prev_sold_date'] = pd.to_datetime(df[['prev_sold_year', 'prev_sold_month', 'prev_sold_day']])
df.drop(['prev_sold_year', 'prev_sold_month', 'prev_sold_day'], axis=1, inplace=True)

# 2. Feature Engineering:

# Extract city and state from address
df[['street_address', 'city_state']] = df['address'].str.split(',', n=1, expand=True).applymap(lambda x: x.strip())
df[['city', 'state']] = df['city_state'].str.split(',', n=1, expand=True).applymap(lambda x: x.strip() if x else '')
df.drop(['address', 'city_state'], axis=1, inplace=True)

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

# 3. Handling Missing Values (if any):

# Check for missing values
print(df.isnull().sum())

# Simple imputation (replace with mean/median or a more sophisticated method)
# Example: df['price'].fillna(df['price'].median(), inplace=True)

# 4. Data Transformation (if needed):

# Example: Convert status to categorical
df['status'] = df['status'].astype('category')

# Display the cleaned and transformed DataFrame
print(df.head())
print(df.info())

# Save to a new CSV file
df.to_csv('cleaned_real_estate_data.csv', index=False)

```

Key improvements and explanations:

- **Data Type Conversion:** Ensures correct data types for numerical calculations and analysis (e.g., price, bed, bath, etc.). Zip code is converted to a string to handle potential leading zeros.
- **Date/Time Handling:** Combines the separate date components into a single `prev_sold_date` column using `pd.to_datetime`. This makes date-based analysis much easier.
- **Address Parsing:** Extracts city and state from the address field for better location-based analysis.
- **Feature Engineering:** Creates a `price_per_sqft` feature, which is often a valuable metric in real estate.
- **Missing Value Handling:** Includes a check for missing values using `df.isnull().sum()`. I've commented out a simple imputation example (using the median). You should choose the most appropriate imputation strategy based on your data and analysis goals.
- **Categorical Conversion:** Converts the `status` column to a categorical type, which is more memory-efficient and can be used directly in some statistical models.
- **Output:** Prints the cleaned data's head and info, and saves it to a new CSV file.

This script provides a solid foundation for data wrangling. You can adapt and extend it based on your specific analysis needs. For example, you might want to add more sophisticated imputation methods, create additional features (e.g., age of the house), or perform more complex data transformations.