I can perform data cleaning on the provided dataset. Here's a step-by-step approach:

Step 1: Import necessary libraries and load the data

```
import pandas as pd
import numpy as np

# Load the data
data = pd.read_csv('data.csv', sep=';')
```

Step 2: Check for missing values

```
# Check for missing values
print(data.isnull().sum())
```

This will print the number of missing values in each column.

Step 3: Handle missing values

```
# Replace missing values in 'street' column with an empty string
data['street'] = data['street'].fillna('')

# Replace missing values in 'city' column with an empty string
data['city'] = data['city'].fillna('')

# Replace missing values in 'state' column with an empty string
data['state'] = data['state'].fillna('')

# Replace missing values in 'zip_code' column with an empty string
data['zip_code'] = data['zip_code'].fillna('')

# Replace missing values in 'house_size' column with 0
data['house_size'] = data['house_size'].fillna(0)

# Replace missing values in 'prev_sold_date' column with an empty string
data['prev_sold_date'] = data['prev_sold_date'].fillna('')
```

Step 4: Check for inconsistent data types

```
# Check for inconsistent data types in 'bed' column
print(data['bed'].unique())

# Check for inconsistent data types in 'bath' column
print(data['bath'].unique())

# Check for inconsistent data types in 'acre_lot' column
print(data['acre_lot'].unique())
```

This will print the unique values in each column, which can help identify inconsistent data types.

Step 5: Handle inconsistent data types

```
# Convert 'bed' column to numeric values
data['bed'] = pd.to_numeric(data['bed'], errors='coerce')

# Convert 'bath' column to numeric values
data['bath'] = pd.to_numeric(data['bath'], errors='coerce')

# Convert 'acre_lot' column to numeric values
data['acre_lot'] = pd.to_numeric(data['acre_lot'], errors='coerce')
```

Step 6: Check for outliers

```
# Check for outliers in 'price' column
print(data['price'].describe())

# Check for outliers in 'house_size' column
print(data['house_size'].describe())
```

This will print the summary statistics for each column, which can help identify outliers.

Step 7: Handle outliers

```
# Remove outliers in 'price' column
data = data[(data['price'] >= 0) & (data['price'] <= 10000000)]

# Remove outliers in 'house_size' column
data = data[(data['house_size'] >= 0) & (data['house_size'] <= 100000)]
```

Step 8: Save the cleaned data

```
# Save the cleaned data to a new CSV file
data.to_csv('cleaned_data.csv', index=False)
```

This will save the cleaned data to a new CSV file named `cleaned_data.csv`.

Here is the complete code:

“‘python import pandas as pd import numpy as np

Load the data

data = pd.read_csv(‘data.csv’, sep=‘;’)

Check for missing values

print(data.isnull().sum())

Replace missing values in ‘street’ column with an empty string

data[‘street’] = data[‘street’].fillna(”)

Replace missing values in ‘city’ column with an empty string

data[‘city’] = data[‘city’].fillna(”)

Replace missing values in ‘state’ column with an empty string

data[‘state’] = data[‘state’].fillna(”)

Replace missing values in ‘zip_code’ column with an empty string

data[‘zip_code’] = data[‘zip_code’].fillna(”)

Replace missing values in ‘house_size’ column with 0

data[‘house_size’] = data[‘house_size’].fillna(0)

Replace missing values in ‘prev_sold_date’ column with an empty string

data[‘prev_sold_date’] = data[‘prev_sold_date’].fillna(”)

Check for inconsistent data types in ‘bed’ column

print(data[‘bed’].unique())

Check for inconsistent data types in ‘bath’ column

print(data[‘bath’].unique())

Check for inconsistent data types in ‘acre_lot’ column

print(data[‘acre_lot’].unique())

Convert ‘bed’ column to numeric values

data[‘bed’] = pd.to_numeric(data[‘bed’], errors=‘coerce’)

Convert ‘bath’ column to numeric values

data[‘bath’] = pd.to_numeric(data[‘bath’], errors=‘coerce’)

Convert ‘acre_lot’ column to numeric values

data[‘acre_lot’] = pd.to_numeric(data[‘acre_lot’], errors=‘coerce’)

Check for outliers in ‘price’ column

print(data[‘price’].describe())

Check for outliers in ‘house_size’ column

print(data[‘house_size’].describe())

Remove outliers in ‘price’ column

data = data[(data[‘price’] >= 0) & (data[‘price’] <= 10000000)]

Remove outliers in ‘house_size’ column

data = data[(data[‘house_size’] >= 0) & (data[‘house_size’] <= 100000)]

Save the cleaned data to a new CSV file

data.to_csv(‘cleaned_data.csv’, index=False)