

Exploring the Nashville Residential Parcel Transaction Dataset

Your Name

2025-10-01

Introduction

Below is introductory code to help introduce you to the `nashville_housing_data.csv` data set, giving you access to sale level data on all residential parcel transaction between 1970 and 2023 in Nashville, Tennessee. It demonstrates basic data cleaning and visualization as a starting point for further exploration!

Load the Dataset

Adjust path to where your dataset is stored. This code utilizes the `tidyverse` and `lubridate` packages, so make sure they are downloaded first!

```
data <- read_csv("nashville_housing_data.csv")
```

Data Cleaning

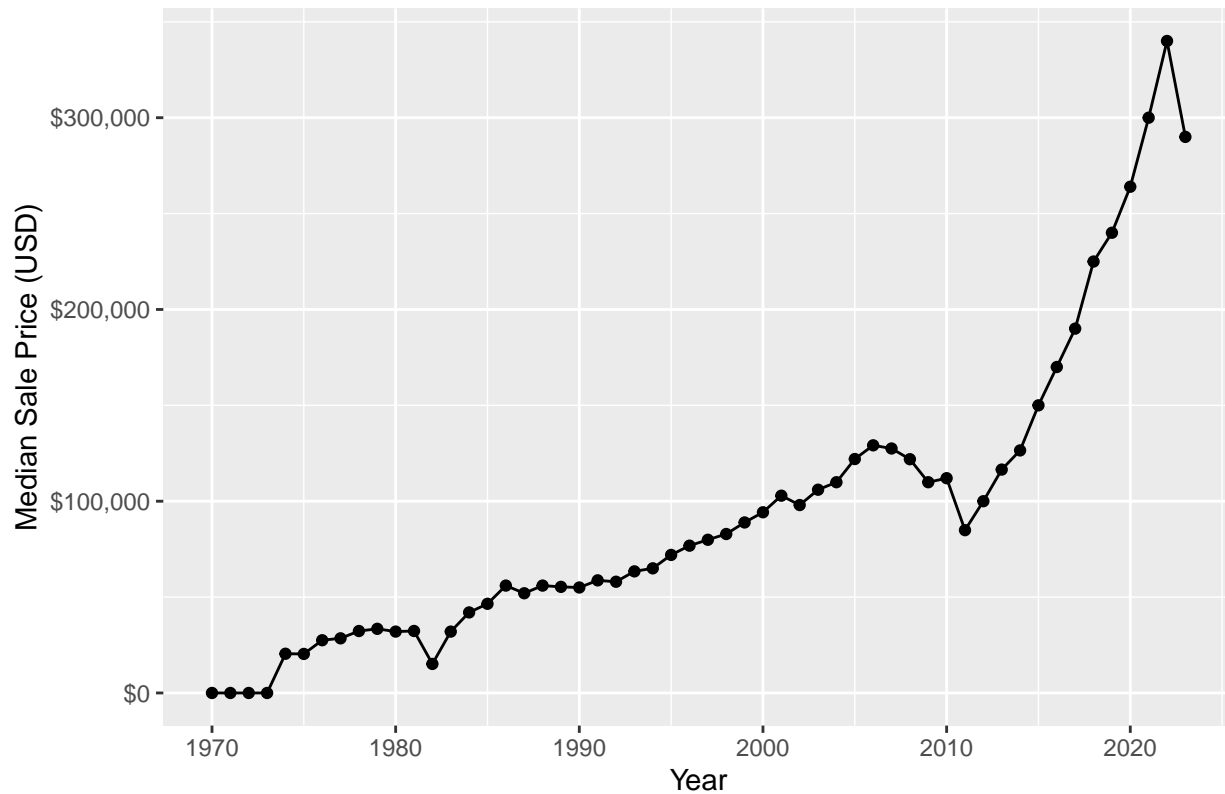
Ensures dates and numeric fields are parsed properly.

```
data <- data %>%  
  mutate(  
    SaleDate = mdy(SaleDate),  
    Year = year(SaleDate),  
    SalePrice = as.numeric(SalePrice),  
    Acres = as.numeric(Acres),  
    FinishArea = as.numeric(FinishArea)  
  )
```

Sale Price Over Time

```
data %>%  
  group_by(Year) %>%  
  summarise(MedianPrice = median(SalePrice, na.rm = TRUE)) %>%  
  ggplot(aes(x = Year, y = MedianPrice)) +  
    geom_line() +  
    geom_point() +  
    labs(  
      title = "Median Residential Sale Price in Nashville (1970-2023)",  
      x = "Year",  
      y = "Median Sale Price (USD)"  
    ) +  
    scale_y_continuous(labels = scales::dollar)
```

Median Residential Sale Price in Nashville (1970–2023)



Time Series Decomposition 2000-2023

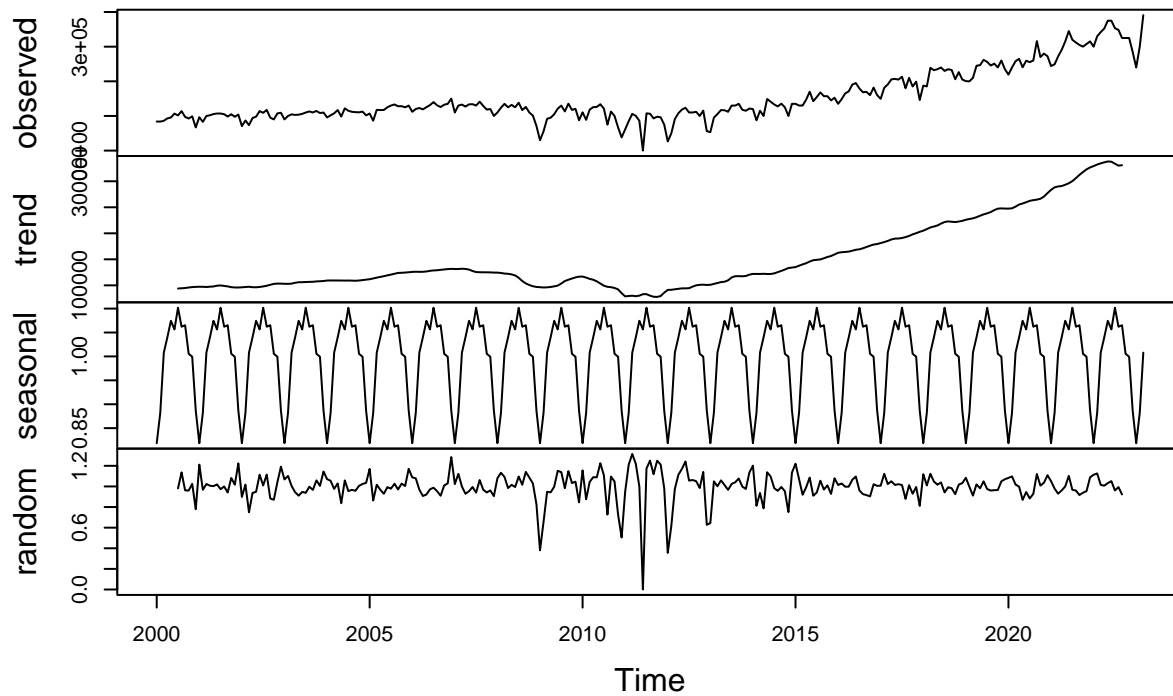
```
data2000 <- data %>%
  filter(Year > 1999) # filter year

# Aggregate to monthly median prices
ts_data <- data2000 %>%
  filter(!is.na(SaleDate), !is.na(SalePrice)) %>%
  mutate(YearMonth = floor_date(SaleDate, "month")) %>%
  group_by(YearMonth) %>%
  summarise(MedianPrice = median(SalePrice, na.rm = TRUE)) %>%
  arrange(YearMonth)

# Create a continuous monthly time series
sale_ts <- ts(
  ts_data$MedianPrice,
  start = c(year(min(ts_data$YearMonth)), month(min(ts_data$YearMonth))),
  end = c(year(max(ts_data$YearMonth)), month(max(ts_data$YearMonth))),
  frequency = 12)

# Decompose
output <- decompose(sale_ts, type = "multiplicative")
plot(output)
```

Decomposition of multiplicative time series



Distribution of Sale Prices 2020-2023

```
data %>%
  filter(SalePrice > 1000, SalePrice < 2000000, Year > 2019) %>% # filter extreme outliers and year
  ggplot(aes(x = SalePrice)) +
  geom_histogram(bins = 50, fill = "maroon", color = "white") +
  labs(
    title = "Distribution of Sale Prices 2020-2023",
    x = "Sale Price (USD)",
    y = "Count"
  ) +
  scale_x_continuous(labels = scales::dollar)
```

Distribution of Sale Prices 2020–2023

