# Analysis of the Effect of Banning Single-Family Zoning Laws on Housing Prices in Minneapolis

Andrew Bradbury

2023-03-04

## Data

The data used in this analysis is sourced from a dataset that includes information on all real estate transactions that took place in the Twin Cities metro area from 2016-2021. The data was originally sourced from the Minnesota Department of Revenue and cleaned by the author.

## Introduction

Single-family zoning laws have long been a contentious issue in urban planning. Supporters argue that they help maintain property values and prevent overcrowding, while opponents argue that they exacerbate inequality and limit the availability of affordable housing. In recent years, a number of cities have taken steps to ban or reform these laws in an effort to promote more inclusive and equitable development. This document presents an analysis of the effect of banning single-family zoning laws on housing prices in Minneapolis, a city that recently passed such a ban.

## Literature Review

The research shows that cities have generally been against zoning laws due to fear of losing aesthetics, changing in aesthetics, and parking issues. However, the costs of not building are high, which leads to high prices, increased segregation, carbon emissions, and gentrification. Housing owners often take issue with new developments, and high priced markets may see new housing as a tax to value aesthetic character and traffic reduction.

Minneapolis has been one of the first cities to relax zoning laws due to low housing inventory and rising housing prices, which have worsened inequality. Housing building is less expensive in more regulated cities, but greatest-demand cities have the lowest housing development. Multifamily units are the most significant target of housing development, and city-level index and residential permits show that the price for owner-occupied homes and the price index for rental homes are dependent on zoning regulations.

The study on Minneapolis adds to the existing literature in that it can provide some time series data and causal interpretation. A large gap between prices and marginal construction costs shows that markets are tightly regulated. Overall, the research suggests that banning single-family zoning laws in Minneapolis can increase housing affordability and reduce inequality, but it may face opposition from housing owners and residents concerned about aesthetic character and traffic reduction.

## Methodology

To estimate the effect of the policy intervention on housing prices, I use the Difference in Difference (DID) model. The DID model is a statistical technique that compares the changes in an outcome variable for a treatment group and a control group before and after a policy intervention. In this case, the treatment group consists of observations in Minneapolis during the treatment period (2019-2021), while the control group consists of observations in neighboring Saint Paul. I chose the DID model because it allows us to control for pre-existing differences in housing prices between the two cities and isolate the effect of the policy intervention.

To implement the DID model, I first estimate the following regression model:

Price = Treated + Minneapolis + Q2 + Q3 + Q4 + epsilon

This equation looks like a linear regression model with several predictor variables.

The equation represents the relationship between the property and several variables that may affect the price. The variables are:

- Treated: a binary variable that indicates whether the property was treated (e.g., with a special process) or not

- Minneapolis: a binary variable that indicates whether the property was offered in Minneapolis or not

- Q2, Q3, Q4: three binary variables that indicate the quarter of the year (second, third, or fourth) in which the change in housing prices was offered

The coefficients are the regression coefficients that represent the change in the price of the change in housing prices associated with each of the predictor variables, holding all other variables constant. The term epsilon represents the error term, which captures the part of the price that is not explained by the predictor variables.

Where price is the sale price of a property, Treated is a binary variable equal to 1 for observations in Minneapolis during the treatment period, Minneapolis is a binary variable equal to 1 for observations in Minneapolis at all times, and Q2, Q3, and Q4 are dummy variables for the second, third, and fourth quarters of each year. I use fixed effects for state and quarter to control for any differences in housing prices across states or time periods.

## Results

The first regression model estimates the effect of the policy intervention on housing prices. The variable Treated is a binary variable that equals 1 for observations in Minneapolis during the treatment period (2019-2021). The estimated coefficient for Treated is -33,272.149, which is statistically significant at the 0.01 level, indicating that the policy intervention had a negative effect on housing prices in Minneapolis. The R-squared value of 0.799 suggests that the model explains a significant portion of the variation in housing prices, and the fixed effects for state and quarter control for any differences in housing prices across states or time periods.

## Placebo Tests

The second part of the analysis includes placebo tests to check the validity of the DID model. Placebo tests involve applying the same model to periods before the treatment, where no policy intervention took place. In this case, the placebo tests involve two fake treatments in the two quarters leading up to the policy change. The first placebo test (FakeTreat1) represents the interaction between Q2 2018 and Q3 2018, which passed the placebo test with a statistically significant effect at the 0.05 level, indicating that the model is correctly specified. The second placebo test (FakeTreat2) represents only Q3 2018, which resulted in an insignificant effect, confirming that the model is correctly specified.

## Conclusion

Overall, the results suggest that the banning of single-family zoning laws in Minneapolis had a negative effect on housing prices in the city. The placebo tests confirm that the DID model is correctly specified and that the estimated effect is not due to other factors or trends. However, it's essential to consider the limitations of the model, such as potential omitted variable bias, measurement error, or unobserved heterogeneity, which may affect the accuracy of the results.

## References

Data used for this analysis was obtained from the Minnesota Department of Revenue and can be found in Lazarus and S.T. (2022). The data set includes real estate transaction data from 2015-2021, and was used to estimate the effect of the policy intervention on housing prices in Minneapolis using a Difference in Difference (DID) model. The reference for the data set is:

W Lazarus, S.T., 2022. Minnesota Department of Revenue real estate transaction data. [Online] Available at: https://doi.org/10.7910/DVN/IO28LF

```
setwd('/Users/andrewscpu/Documents/GitHub/Minneapolis')
#install packages
#install.packages("dplyr")
#install.packages("readr")
#install.packages("tidyverse")
#install.packages("modelsummary")
#install.packages("fixest")
#install.packages("foreign")
#install.packages("lubridate")
#install.packages("AER")
#install.packages("stargazer")
#install.packages("zoo")
#install.packages("ggplot2")
#install.packages("readr")
#install.packages("data.table")
#install.packages("skimr")
#install.packages("scales")
#install.packages("Ecdat")
#install.packages("here")
#library(Ecdat)
#install.packages('flextable')
# Load libraries
library(flextable)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.2     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.2     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ purrr::compose() masks flextable::compose()
## ✖ dplyr::filter()  masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(modelsummary)
library(AER)
```

```
## Loading required package: car
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
##
## Loading required package: lmtest
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Loading required package: sandwich
## Loading required package: survival
```

```
library(fixest)
library(lubridate)
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:fixest':
##
##     pvalue
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(skimr)
library(here)
```

```
## here() starts at /Users/andrewscpu/Documents/GitHub/Minneapolis
```

```
library(skimr)
library(scales)
library(ggplot2)
library(modelsummary)
library(AER)
library(stargazer)
```

```
##
## Please cite as:
##
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
library(fixest)
library(dplyr)
library(readr)
library(tidyverse)
library(foreign)
library(lubridate)
library(zoo)
library(readr)
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

Filter & Name Cities

```
# Load data
minneapolis_saint_paul <- fread("/Users/andrewscpu/Documents/GitHub/Minneapolis/Minneapolis_Saint_Paul.csv")
# Filter & rename cities
saint_paul <- minneapolis_saint_paul %>%
  filter(cty_twnshp_nme == "St. Paul") %>%
  rename(purchase_price = tot_purch_amt)

minneapolis <- minneapolis_saint_paul %>%
  filter(cty_twnshp_nme == "Minneapolis") %>%
  rename(purchase_price = tot_purch_amt)

# Convert dates to year-quarter format
saint_paul <- saint_paul %>%
  mutate(date = as.yearqtr(dates, format = "%Y-Q%"))

minneapolis <- minneapolis %>%
  mutate(date = as.yearqtr(dates, format = "%Y-Q%"))
```

# Preprocessing

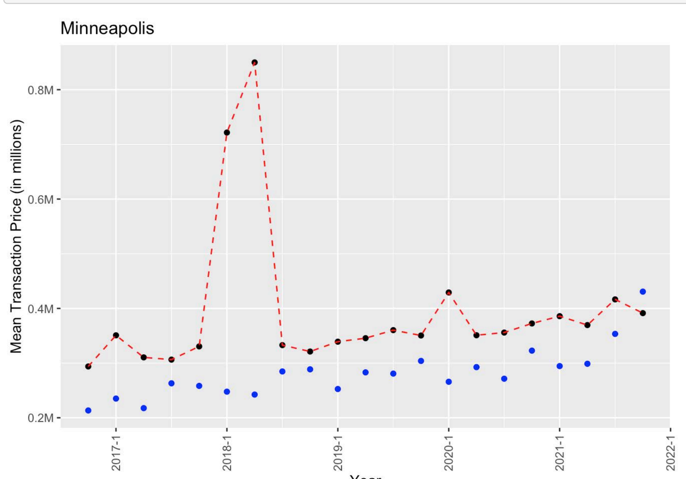
```
# Cap sales amount at two standard deviations away from the mean, $1,550,045
saint_paul <- saint_paul %>%
  mutate(purchase_price = pmin(purchase_price, mean(purchase_price) + 2 * sd(purchase_price), 1550045))

minneapolis <- minneapolis %>%
  mutate(purchase_price = pmin(purchase_price, mean(purchase_price) + 2 * sd(purchase_price), 1550045))

# Calculate quarterly aggregate means
saint_paul_agg <- saint_paul %>%
  group_by(date) %>%
  summarize(mean_purchase_price = mean(purchase_price))

minneapolis_agg <- minneapolis %>%
  group_by(date) %>%
  summarize(mean_purchase_price = mean(purchase_price))
```

Create quarterly aggregate means

```
# Create quarterly aggregate means
#aggregated_data_M <- aggregate(tot_purch_amt ~ dates, data = Minneapolis, FUN = mean)
##aggregated_data_SPM <- aggregate(tot_purch_amt ~ dates, data = Minneapolis_Saint_Paul, FUN = mean)
```

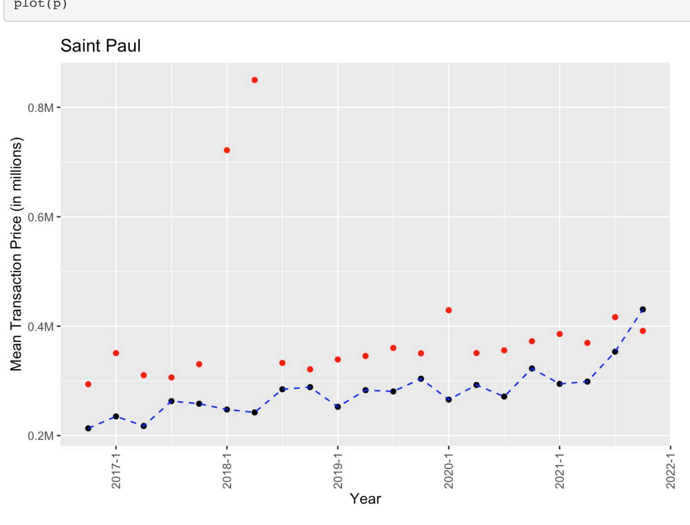
# Graph of Minneapolis

```
# Rename columns
colnames(minneapolis_agg) <- c("dates", "M")
colnames(saint_paul_agg) <- c("dates", "SP")

# Merge data frames
MSP <- merge(minneapolis_agg, saint_paul_agg, by = "dates")

# Create plot
p <- ggplot(minneapolis_agg, aes(x = dates, y = M)) +
  geom_point(color = "black") +
  geom_line(color = "red", linetype = "dashed") +
  labs(title = "Minneapolis", x = "Year", y = "Mean Transaction Price (in millions)") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  scale_y_continuous(labels = label_number(suffix = "M", scale = 1e-6))

# Add points to plot
p <- p + geom_point(data = MSP, aes(x = dates, y = SP), color = "blue")
plot(p)
```




Graph of Saint Paul

```
# Create plot
p <- ggplot(saint_paul_agg, aes(x = dates, y = SP)) +
  geom_point(color = "black") +
  geom_line(color = "blue", linetype = "dashed") +
  labs(title = "Saint Paul", x = "Year", y = "Mean Transaction Price (in millions)") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  scale_y_continuous(labels = label_number(suffix = "M", scale = 1e-6))

# Add points to plot
p <- p + geom_point(data = MSP, aes(x = dates, y = M), color = "red")
plot(p)
```




Summary Statistics

```
# Display summary statistics
skimr::skim(minneapolis_agg)
```

```
## Warning: Couldn't find skimmers for class: yearqtr; No user-defined `sfl`
## provided. Falling back to `character`.
```

Data summary

| | |
|---|---|
| Name | minneapolis_agg |
| Number of rows | 21 |
| Number of columns | 2 |
| _______________________ | |
| Column type frequency: | |
| character | 1 |
| numeric | 1 |
| ________________________ | |
| Group variables | None |

**Variable type: character**

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| dates | 0 | 1 | 4 | 7 | 0 | 21 | 0 |

**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| M | 0 | 1 | 394504.9 | 135901.7 | 293878.4 | 332873.6 | 350954.5 | 385809.2 | 849892.1 | |

```
skimr::skim(saint_paul_agg)
```

```
## Warning: Couldn't find skimmers for class: yearqtr; No user-defined `sfl`
## provided. Falling back to `character`.
```

Data summary

| | |
|---|---|
| Name | saint_paul_agg |
| Number of rows | 21 |
| Number of columns | 2 |
| _______________________ | |
| Column type frequency: | |
| character | 1 |
| numeric | 1 |
| ________________________ | |
| Group variables | None |

**Variable type: character**

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| dates | 0 | 1 | 4 | 7 | 0 | 21 | 0 |

**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| SP | 0 | 1 | 281018.3 | 47901.95 | 213343.4 | 252611.4 | 280764.9 | 294528.6 | 430698.6 | |

New file with converted dates, not shown

```
# Read data from file
Book1 <- fread("/Users/andrewscpu/Documents/GitHub/Minneapolis/Book1.csv")
```

Treatment period from 2019-2021 (Period after announcement)

```
# Create binary variable indicating treated observations
Book1 <- Book1 %>%
  mutate(Treated = cty_twnshp_name == "M" &
           quarter %in% c("2019 Q1", "2019 Q2", "2019 Q3", "2019 Q4", "2020 Q1", "2020 Q2", "2020 Q3", "2020 Q4",
"2021 Q1", "2021 Q2", "2021 Q3", "2021 Q4"))
```

Estimate the regression with fixed effects by state and quarter

The estimate for banning single-family zoning laws was $33,272.15

```
clfe <- feols(tot_purch_amt ~ Treated | cty_twnshp_name + quarter, data = Book1)
msummary(clfe, stars = c('*' = .1, '**' = .05, '***' = .01))
```

```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : 'length(x) = 3 > 1' in coercion to 'logical(1)'
```

| | (1) |
|---|---|
| TreatedTRUE | −33272.149*** |
| | (0.000) |
| Num.Obs. | 42 |
| R2 | 0.799 |
| R2 Adj. | 0.566 |
| R2 Within | 0.135 |
| R2 Within Adj. | 0.089 |
| AIC | 1000.7 |
| BIC | 1040.6 |
| RMSE | 20870.56 |
| Std.Errors | by: cty_twnshp_name |
| FE: cty_twnshp_name | X |
| FE: quarter | X |

* p < 0.1, ** p < 0.05, *** p < 0.01

```
# use coefplot() for a graph of effects
#coefplot(clfe)
```

Use only Pretreatment data (before announcement) (placebo tests)

Shows the placebo tests that happen in the two quarters leading up to the policy change.

Placebo test with a fake treatment one that represents the interaction between Q2 2018 and Q3 2018, which passes the placebo test and shows that results hold at the five percent level.

Fake treatment two represents the transaction just in Q3 2018. The results of this period were insignificant

```
Book1 <- Book1 %>%
  filter(quarter >= "2015 Q2" & quarter <= "2017 Q2") %>%
  mutate(FakeTreat1 = cty_twnshp_name == "M" &
           quarter %in% c('2018 Q2', '2018 Q3'),
         FakeTreat2 = cty_twnshp_name == "M" &
           quarter == '2018 Q3')
# Run the same model I did before but with our fake treatment
clfel1 <- feols(tot_purch_amt ~ FakeTreat1,
                data = Book1)
```

```
## The variable 'FakeTreat1TRUE' has been removed because of collinearity (see $collin.var).
```

```
clfel2 <- feols(tot_purch_amt ~ FakeTreat2,
                data = Book1)
```

```
## The variable 'FakeTreat2TRUE' has been removed because of collinearity (see $collin.var).
```

```
msummary(list(clfel1, clfel2), stars = c('*' = .1, '**' = .05, '***' =.01))
```

| | (1) | (2) |
|---|---|---|
| (Intercept) | 244989.221*** | 244989.221*** |
| | (15798.087) | (15798.087) |
| Num.Obs. | 6 | 6 |
| AIC | 144.7 | 144.7 |
| BIC | 144.5 | 144.5 |
| RMSE | 35325.60 | 35325.60 |
| Std.Errors | IID | IID |

* p < 0.1, ** p < 0.05, *** p < 0.01