Impact of right to repair law in Massachusetts' repair markets

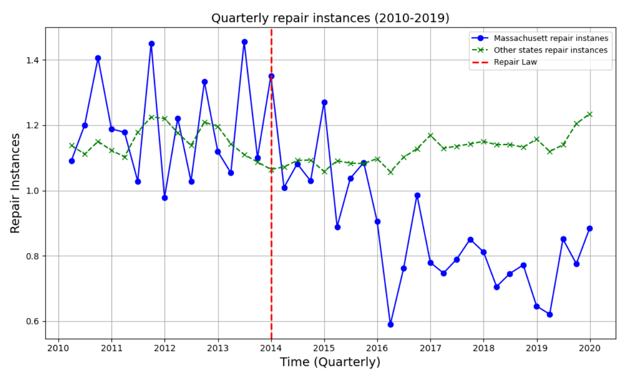
Minjae Seo

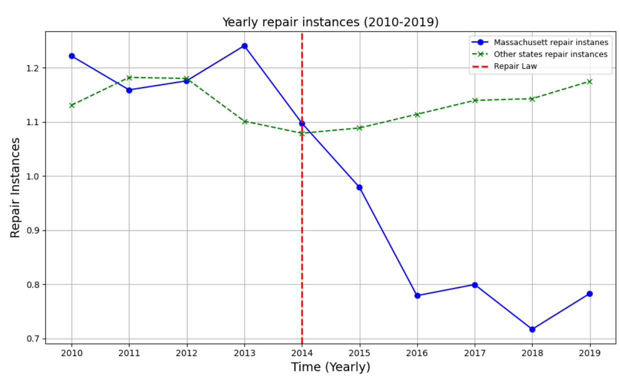
June 2, 2024

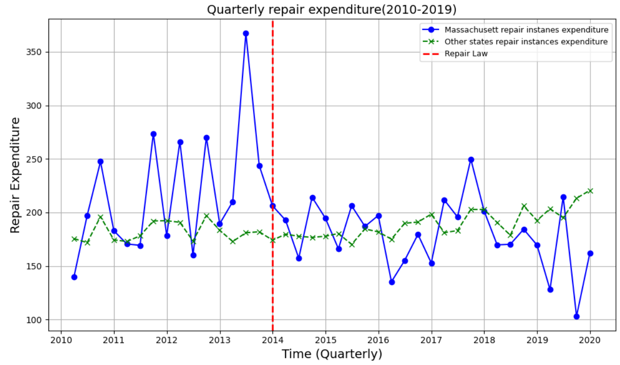
Introduction

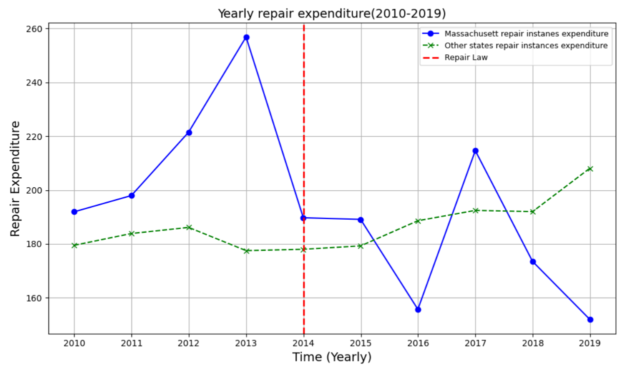
We want to examine how the right-to-repair law in the 2004 repair market affects Massachusetts (the treatment group) and the control group (other states) and their significance. In my exploratory data analysis (EDA) and causal analysis, I used two outcome variables: VOPMOA, which records the repair month for each unique household, and VOPXEXPX, which details each household's repair expenditures. Additionally, I created a new variable based on the previous two to assess the total repair expenses for each household.

Exploratory Data Analysis









Yearly repair instances after the right to repair raw(after 2014) in Massachusetts decreased significantly compared to other states and quarterly repair instances. There's also a parallel trend between treated and controlled groups, but this isn't perfect due to the randomness or latent variables(Omitted variables) we can't observe. For repair expenditure, there is a more clear parallel trend between treated and controlled groups except for the economic shock from 2015 to 2016. Also, we couldn't just say that the impact of raw is significant on repair expenditure in Massachusetts. Also, there was a substantial decrease in repair expenditure in MA between 2013 and 2014, just before the intervention. We will run a difference in difference regression to quantify the estimated treatment effect and differential effect between Massachusetts and other states. And, I will explain the exact regression equation and assumption in the methodology section.

Methodology

$$Y_{it} = \beta_0 + \beta_1 \times treatment_i + \beta 2 \times post_t + \beta_3 \times (post_t \times treatment_i) + \epsilon_{it}$$

- β_0: Expected repair instances for control states before the right-to-repair law.
- β_1: Expected differences in repair instances between treatment(MA) and control(Other states) groups.
- β_2: Expected differences in repair instances between Massachusetts and other states before and after the right-to-repair law.
- β_3: Expected difference-in-difference between Massachusetts and control states with the difference in before and after the raw.
- ϵ_{it}: random noise and the unobserved factor.

Causal Analysis

Differential treatment effect

Dep. Variable:	REPAIR_INSTANCES	**R-squared:**	0.001
Model:	OLS	**Adj. R-squared:**	0.001
Method:	Least Squares	**F-statistic:**	60.80
Date:	Fri, 10 May 2024	**Prob (F-statistic):**	2.74e-39
Time:	20:24:39	**Log-Likelihood:**	-2.2716e+05
No. Observations:	357028	**AIC:**	4.543e+05
Df Residuals:	357024	**BIC:**	4.544e+05
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	0.7075	0.001	597.743	0.000	0.705	0.710
TREATMENT	0.0194	0.008	2.460	0.014	0.004	0.035
POST	-0.0079	0.002	-5.033	0.000	-0.011	-0.005
TREATMENT:POST	-0.0996	0.010	-9.489	0.000	-0.120	-0.079

Omnibus:	407102.349	**Durbin-Watson:**	1.482
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	68488.189
Skew:	-0.884	**Prob(JB):**	0.00
Kurtosis:	1.784	**Cond. No.**	19.1

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

These difference-in-difference regression results represent the impact of law on repair law in Massachusetts. Based on the p-value and significance, we can conclude that the legislation has a significant treatment effect. However, I would also like to see whether the other states also had treatment effects.

Placebo-test

Based on the Placebo test, we can see less treatment effect in other states. However, some of the states still have differential treatment effects, therefore we can't just conclude it is a causal effect.

Lead-Lag Analysis

Based on the Lead-Lag analysis, we can see that except for 2011, adjusting the baseline year as earlier also has a differential treatment effect.

Code

```
data = final_merge.copy()
data['YEAR'] = data['YEAR'] // 10
data['POST'] = (data['YEAR'] >= 2014).astype(int)
data['TREATMENT'] = (data['STATE'] == 25).astype(int)
data['TREATMENT:POST'] = data['TREATMENT'] * data['POST']

model = 'REPAIR_INSTANCES ~ TREATMENT + POST + TREATMENT:POST'
fitted_model = smf.ols(model, data=data).fit()
print("Regression Results in MA:")
print(fitted_model.summary())
```

Listing 1: DID-Analysis

```python
def placebo_tests(data, test_states):
    # Prepare a dictionary to store results for each test state
    results = {}

    # Preprocess data only once
    data['YEAR'] = data['YEAR'] // 10
    data['POST'] = (data['YEAR'] >= 2014).astype(int)

    # Iterate over each state in the list of test states
    for test_state in test_states:
        data['TREATMENT'] = (data['STATE_NAME'] == test_state).astype(int)
        data['TREATMENT:POST'] = data['TREATMENT'] * data['POST']

        model = 'REPAIR_INSTANCES ~ TREATMENT + POST + TREATMENT:POST'
        fitted_model = smf.ols(model, data=data).fit()
        results[test_state] = fitted_model.summary()

    return results

final_merge_did = final_merge.copy()
test_states = final_merge_did['STATE_NAME'].unique()
placebo_results = placebo_tests(final_merge_did, test_states)

for state, result in placebo_results.items():
    print(f"Placebo test results for {state}:")
    print(result)
```

Listing 2: Placebo test

```python
def lead_lag_analysis(data, test_state='Massachusetts'):
    data['YEAR'] = data['YEAR'] // 10
    years = list(range(2010, 2020))
    results = {}
    confidence_intervals = {}
    standard_errors = {}

    for year in years:
        data['POST'] = (data['YEAR'] >= year).astype(int)
        data['TREATMENT'] = (data['STATE_NAME'] == test_state).astype(int)
        data['TREATMENT:POST'] = data['TREATMENT'] * data['POST']

        model = 'REPAIR_INSTANCES ~ TREATMENT + POST + TREATMENT:POST'
        fitted_model = smf.ols(model, data=data).fit()

        results[year] = fitted_model.summary()
        confidence_intervals[year] = fitted_model.conf_int()
        standard_errors[year] = round(fitted_model.bse, 3)

    return results, confidence_intervals, standard_errors

placebo_results, ci_results, se_results = lead_lag_analysis(final_merge.copy())

for year in se_results:
```

```python
print(f"Regression Results {year}:")
print(placebo_results[year])
```

Listing 3: Lead-Lag analysis

Links to code: Code