EC413 Applied Econometrics Assignment 2:

Difference in Differences

Measuring Effectiveness of The 4pm Off-premises Alcohol Sale Ban

Student Number: 202133230

Word Count: 1493

Introduction

This report aims to measure and discuss the policy effect of the ban of alcohol sales at offsite premises after 4pm in efforts to reduce alcohol related hospitalisations (ARH) across the cities of Natural Experiments.

Data Discussion

The policy was not adopted simultaneously for all cities shown by Figure 1. 3 cities were very early adopters, where the policy came into effect in 2013. Conversely the majority adopted in 2022, and the remainder in 2017. There are alternative methods for addressing a staggered rollout but this report, constrained by time horizons of the data provided of 2013-2022, omits the early adopters and cuts the analysis period to 2013-2021, reducing the number of cities from 50 to 47 and the total number of observations by 15% from 500 to 423, using the late adopters as a control group. This is because measuring the pre policy trends of the early adopters and the post policy trends on the late is not feasible due to insufficient data. This has the added benefit of avoiding the problem of heterogeneous treatment effects cause by early adopters receiving the policy effect while others in the treatment group are still in their pre policy periods, effectively analysing the staggered rollout as a single start with the 2017 policy start group as the treatment group.

Assessing the Common Trends Assumption

In a Difference-in-Differences (DiD) analysis, the common trends assumption (CTA) states that in the absence of the policy, the treated and control group would have followed similar paths over time such that the trend of the observed control group can be used to estimate the trend of the unobserved counterfactual; effect of no treatment on the treatment group, the difference between which is the average treatment effect on the treated i.e. the true policy effect. If violated, the resulting DiD estimator would be biased, incorporating both the policy effect as well as the trend not common between the treatment and control groups.

A visual check was employed to test whether this key assumption holds by plotting a simple line graph of the ARH of the treatment and control groups shown in Figure 2 with a vertical line demarcating the policy start. Evidently both groups follow a common upward trend in the ARH during the pre-policy period, supporting the assumption of common trends. This is further supported by the pre-policy event study coefficient estimates being insignificantly different from zero in Figure 3's event study plot that will be discussed in further detail.

Model Specification:

1. $ARH_{ct} = a + \beta_1 Treatment_c + \beta_2 Post_t + \delta(Treatment_c \times Post_t) + \varepsilon_{ct}$

2. $ARH_{ct} = a + \sum_{k=1}^{46} \beta_k city_k + \sum_{j=2014}^{2021} y_j year_j + \delta(Policy\_in\_Effect)_{ct} + \varepsilon_{ct}$

The final specification is derived from the general form seen in equation 1. It is more appropriate to use the extended, Two-Way Fixed Effects (TWFE) form in equation 2 as it allows for each year and city to have its own linear trend relative to a reference period as opposed to a single generalised coefficient, resulting in more tailored isolation of the common trend and accurate estimation of the policy effect. Because CTA holds, both models provide the same coefficient estimates.

As made evident by Figure 7 , not all cities start from the same level of alcohol related hospitalisations level that may be due to several different factors. The city fixed effects account for any level differences in the cities that stay constant over the course of the analysis window such as cultures effect on drinking rates. The year fixed effects account for any time varying changes across all the cites e.g. nation-wide tariff on alcohol in 2014. The alpha parameter is the intercept. Representing the average level of alcohol related hospitalisations per 10k in the control group. The sigma coefficient or difference in differences estimator is the key estimate in the model that, if CTA holds, estimates the true causal policy effect by measuring the marginal effect of being in the treatment group, after treatment on the ARH value.

<u>Concerns</u>

This methodology also assumes independence in the errors across observations i.e. treatment/ policy is assigned randomly. But since policy adoption year is unlikely to be random due to factors involved in passing and implementing a policy, there are likely shared characteristics between the 2017 adopters that cause them to covary. The intuition of which is discussed in the Results & Interpretation section. As a result the variance within groups tends to be underestimated, increasing the risk of Type 1 error where the report falsely rejects the null that there is no policy effect and concludes there is a significant one due to narrow standard error estimates. For robustness the standard errors were clustered on a national level, the results of which are shown in column 2 of Figures 4 and 5.

This model also assumes there are no spillover effects i.e. effects of policy or no policy affects other cities. There is no strong intuition for spillover effects as it is unlikely citizens will relocate to a different city due to this policy. Though a study into intercity travel for day trips or nights-out to have a conclusive answer could be done for further rigour.

Lack of controls is another limitation to be concerned about as exogenous variables such as other alcohol restricting policies would make the analysis more robust.

Results, Interpretation & Intuition

As shown in Figures even with the robust standard errors the DiD estimator returns a policy effect of -5.32 with a standard error of 0.72, statistically significant at a less than 1% level. This means the implementation of the policy is associated with a fall in the number of alcohol related hospitalisations by 5.32 per 10 thousand of the population . Since the CTA stands, as defended by the above this can be interpreted as a causal effect of the policy. Comparing this to the intercept from the model of 77.325, this is a fall of 6.88%. Though this is a significant result, it is relatively small. Likely because the ban comes into effect very early in the day when alcohol related hospitalisations tend to occur later in the night. People may opt to just buy their drinks earlier.

As mentioned, there are concerns around the independence of errors within groups which are intensified by Figure 2's plot. It is shown that the treatment group (2017 adopters), though following the same trend as the control, have a consistently higher ARH. It is likely that the cities that had the biggest issues with ARH would be quicker to adopt the policy to address it, and cities where it is less of an issue opt to differ. This relates to the concern of control also as if there is strong intuition for treatment not being assigned randomly, bias can be re-introduced e.g. heterogenous treatment effects depending on ARH level that should be controlled for.

It is important to look out for early effects of the treatment on the treatment group as it implies actors change their behaviour in anticipation of the policy start. This is called an anticipation effect and can bias the estimate in either direction depending on the direction and significance. Though the treatment group see a fall in the number of ARH per 10k of population in 2016, implying there is a change in behaviour in anticipation of the 2017 policy start, the fall is not significantly different from zero shown by Figure 8, putting to rest the concern.

Event Study

$$y_{ct} = a + \sum_{j=2}^{4} \beta_j (Lead\ j)_{ct} + \sum_{k=0}^{3} \delta_k (Lag\ k)_{ct} + \mu_c + \gamma_t + \varepsilon_{ct}$$

An event study was carried out and coefficients plotted in Figures 3 and 8. Its coefficients are used to visualise how the treatment group's trend differs from the estimated common trend at various times before and after the policy start relative to a reference period which is primarily 2016. Though Figure 8 uses lead 2, i.e. 2015 as the reference to illustrate an earlier point. In the pretreatment period the estimates are insignificantly different from zero but following treatment, fall below zero, lending credence to both CTA and the significance of the policy effect. Due to the short time horizon provided by the data, the full lead and lag lengths available were used. The specification is given by the city and year fixed effects of mew and gamma respectively, and a lead event study parameter for each year prior and lag for each year following the treatment.

Conclusion

To conclude this report has found a significant effect of the 4pm off premises alcohol sale ban among the cities of Natural Experiments. And through the standing of the CTA it can be interpreted as a causal effect of 5.32 fall in the alcohol related hospitalisations per 10k of population. Though it is significant thus effective, its effect of only 6.88% is small. Though there were a few concerns regarding the lack of controls for other alcohol related policies that would provide depth to the understanding of how the policy works, this report still presents a strong high-level overview of the policy effect.

Appendix:

Figure 2: Alcohol Related Hospitalisations Over Time

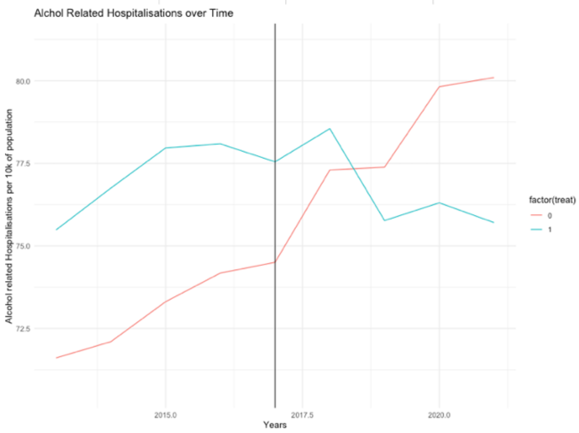

Figure 3: Event Study Coefficients Over Event Time (Lead 1 as reference)

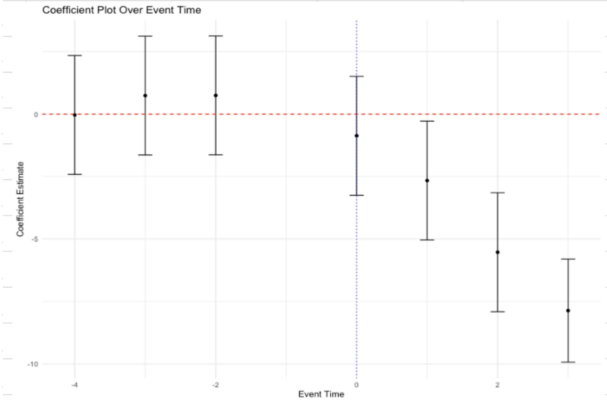

Figure 7: Alcohol Related Hospitalisations By City Over Time

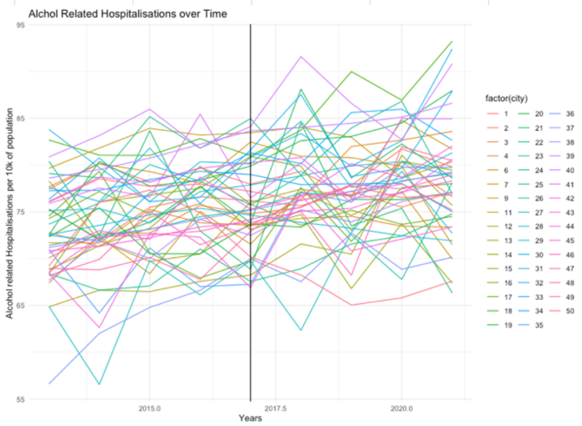

Figure 8: Event Study Coefficients Over Event Time (Lead 2 as reference)

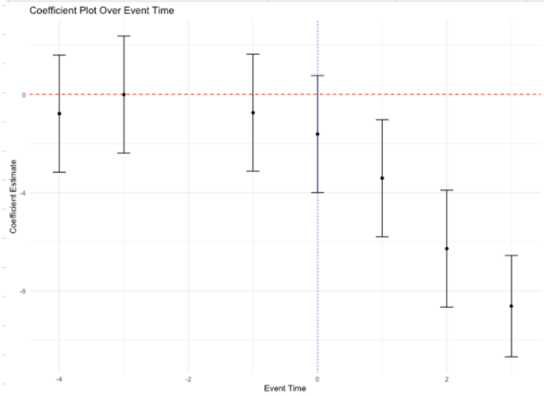

Figure 1: Policy Adoption Year Table

Treatment Start	Number of Cities Adopted
2013	3
2017	18
2022	29

Figure 4: Brief Summaries

	1	2
Effect	-5.320	-5.320
Standar Error	(0.61)	(0.76)
Significance	<0.1%	<0.1%

Figure 5: DiD Regression Summary

	Estimate	Std..Error	t.value	Pr...t..
(Intercept)	77.3265	1.1169	69.2317	9.54E-213
factor(year)2014	0.7878	0.6276	1.2553	2.10E-01
factor(year)2015	1.9998	0.6276	3.1865	1.56E-03
factor(year)2016	2.5829	0.6276	4.1156	4.77E-05
factor(year)2017	4.6124	0.6700	6.8845	2.52E-11
factor(year)2018	6.7213	0.6700	10.0323	4.32E-21
factor(year)2019	5.7091	0.6700	8.5214	4.14E-16
factor(year)2020	7.4186	0.6700	11.0730	8.81E-25
factor(year)2021	7.3594	0.6700	10.9847	1.84E-24
D	-5.3196	0.6124	-8.6867	1.25E-16

Figure 6: Robust Standard Error DiD Regression Summary

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	77.3265	0.5187	149.0817	0.00E+00
factor(year)2014	0.7878	0.5928	1.3289	1.85E-01
factor(year)2015	1.9998	0.5482	3.6476	3.03E-04
factor(year)2016	2.5829	0.6404	4.0333	6.69E-05
factor(year)2017	4.6124	0.7403	6.2308	1.27E-09
factor(year)2018	6.7213	0.5940	11.3162	1.14E-25
factor(year)2019	5.7091	0.6451	8.8502	3.76E-17
factor(year)2020	7.4186	0.6601	11.2382	2.20E-25
factor(year)2021	7.3594	0.6962	10.5705	5.63E-23
D	-5.3196	0.7633	-6.9693	1.48E-11

```r
library(tidyverse)
library(AER)
library(sandwich)
library(lmtest)
library(ggplot2)
library(modelsummary)

options(scipen =4)

### Load data + Clean Data #################################
setwd('/Users/ifechiekekwe/Documents/Uni work/4th Year/Econ/Econometrics/Topic 2 Labs/Assignment 2 (DiD Topic)-20241029')
data <- read.csv('drinkhh_city_panel.csv')
data <- subset(data, year != 2022)

  ### Adoption Year Table ###############
treatment_counts <- data %>%
  group_by(treatment_start) %>%
  summarize(cities_adopted = n_distinct(city)) %>%
  arrange(treatment_start)
# Print the table
print(treatment_counts)

data <- subset(data, data$treatment_start > 2014)#zero observations dropped. Data Set is complete with no missing values
view(data)
### Setting up variables and Descriptive Stats ##########################

data$treat <- ifelse(data$treatment_start ==2017,1,0)

data$post <- ifelse(data$year >=2017,1,0)

data$D <- ifelse(data$treatment_start == 2017 & data$year >= 2017, 1,0)

### Plot for intuition + Visual check of common Trends assumption #########
ggplot(data,
       aes(x = year, y = drinkhh, colour = factor(treat), group = factor(treat))) +
  stat_summary(geom = 'line') +
  geom_vline(xintercept = 2017) +
  labs(
    title = "Alchol Related Hospitalisations over Time",
    x = "Years",
    y = "Alcohol related Hospitalisations per 10k of population"
  ) +
  theme_minimal()

### Regressions  #############################
First <- lm(drinkhh ~ treat + post + treat*post, data = data)
summary(First)

Second <- lm(drinkhh ~ factor(city) + factor(year) + D, data = data)
summary(Second)
### The above should be the same because we are just allowing for each city to have its own linear coefficient

# allowing for unobserved linear variations across cities. (Group Linear Trends)
Third <- lm(drinkhh ~ year*factor(city) + factor(year) + factor(city) + D, data = data)
summary(Third)
### Too intensive on the data and removes almost all other variation
```

```
##########################################################################
### Final Specification + Robustness Check:
##########################################################################
Final <- lm(drinkhh ~ factor(city) + factor(year) + D, data = data)
Final_coefs <- data.frame(summary(Final)$coefficients)
coi_indices <- which(!startsWith(row.names(Final_coefs),'factor(city)'))
FinalShort <- Final_coefs[coi_indices,]
FinalShort

# Robust Standard errors (clustering cities on a national level)
Final_R <- coeftest(Final, vcov = vcovCL, cluster =~city)
FinalShort_R <- Final_R[coi_indices,]
FinalShort_R
##########################################################################
### Event Study ###################
    # Create Leads #####
j = 2017
for(k in 0:4) {
  # assign leads 0-4
  print(j)
  if(k < 4) {
    data[[paste0("Lead_",k)]] <- ifelse(data$treat==1 & data$year==j, 1,0)
  }
# assign last lead
  else {
    data[[paste0("Lead_",k)]] <- ifelse(data$treat==1 & data$year<=j, 1,0)
  }
  j = j - 1
}
```

```r
# Define a numeric event time variable aligned with this specific order
# (e.g., Furthest lead -> Event time <- Furthest lag)
event_time <- c(-4, -3, -2, -1, 0, 1, 2, 3)

# Create Event Study df
ES_df <- data_frame(
  event_time = event_time,
  coefficient = event_coeffs_ordered,
  std_error = event_se_ordered
)

# Add confidence intervals
ES_df$ci_upper <- ES_df$coefficient + 1.96 * ES_df$std_error
ES_df$ci_lower <- ES_df$coefficient - 1.96 * ES_df$std_error

# Plot using ggplot2
ggplot(ES_df, aes(x = event_time, y = coefficient)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_lower, ymax = ci_upper), width = 0.2) +
  labs(x = "Event Time", y = "Coefficient Estimate",
       title = "Coefficient Plot Over Event Time") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +    # Add horizontal line at y = 0
  geom_vline(xintercept = -0, linetype = "dotted", color = "blue") +   # Add vertical line just before event time 0 (e.g., at x = -0.5)
  theme_minimal()

###################################################################
mean(data$drinkhh[data$treatment_start == 2017 & data$year <= 2017])

ggplot(data,
       aes(x = year, y = drinkhh, colour = factor(city))) +
  stat_summary(geom = 'line') +
  geom_vline(xintercept = 2017) +
  labs(
    title = "Alchol Related Hospitalisations over Time",
    x = "Years",
    y = "Alcohol related Hospitalisations per 10k of population"
  ) +
  theme_minimal()
```

```r
    # Create Lags ####
j=2018
for(k in 1:3) {
  if(k < 3) {
  print(j)
  # assign lags 1-4
  data[[paste0("Lag_",k)]] <- ifelse(data$treat==1 & data$year==j, 1, 0)
}
#assign last lag
else {
    data[[paste0("Lag_",k)]] <- ifelse(data$treat==1 & data$year>=j, 1,0)
  }
  j = j + 1
}

  # Run event study ######
eventdd <- lm(drinkhh ~ Lead_4 + Lead_3 + Lead_2 + Lead_0 + Lag_1 + Lag_2 + Lag_3
                + factor(year) + factor(city), data=data)
# don't report the group FE coefs to declutter results window
event_coefs <- data.frame(summary(eventdd)$coefficients)
coi_indices <- which(!startsWith(row.names(event_coefs), 'factor(city)'))

# Extract all coefficients and standard errors
event_coeffs <- coef(eventdd)
event_se <- sqrt(diag(vcov(eventdd)))

# Filter only the coefficients starting with "Lead_" or "Lag_"
event_se_filtered <- event_se[grep("Lead_\\d+|Lag_\\d+", names(event_se))]
event_coeffs_filtered <- event_coeffs[grep("Lead_\\d+|Lag_\\d+", names(event_coeffs))]

# Extract event time indicators as specified by the order
event_time_names <- names(event_coeffs_filtered)

# Manually define event time ordering by
# Creating a vector to specify the order
event_time_order <- c("Lead_4", "Lead_3", "Lead_2", "Lead_1", "Lead_0",
                      "Lag_1", "Lag_2", "Lag_3")

# Reorder coefficients and standard errors according to event_time_order
event_coeffs_ordered <- event_coeffs_filtered[event_time_order]
event_se_ordered <- event_se_filtered[event_time_order]
```