Senior Project

Department of Economics

Expanding State Tax Authority: Public and Private Sector Impacts of *Wayfair*

Maxwell V. Pastoria

Spring 2025

Advisor: Dr. Ali Enami

Abstract

The 2018 Supreme Court decision in *South Dakota v. Wayfair, Inc.* (*Wayfair*) overturned the longstanding physical presence rule for state sales tax collection, enabling states to enforce tax obligations on remote sellers through marketplace facilitator and economic nexus laws (Oyez, 2018). This study investigates the effects of these two post-*Wayfair* tax laws on state sales tax revenues, entrepreneurship, and public expenditures across all 50 states between 2008 and 2023. Using a two-way fixed effects difference-in-differences (TWFE DID) framework, this research analyzes how policy adoption influenced per-capita outcomes in tax revenue, firm formation, and government spending in social services, education, infrastructure, and security. The findings indicate that marketplace facilitator and economic nexus laws did not significantly change state sales tax revenues or firm establishment rates for small businesses (1 to 4 employees). However, the adoption of marketplace facilitator laws is associated with a statistically significant decline in per-capita social services spending. Control variables such as state sales tax rates, GDP, and personal income were consistently significant predictors of both revenue and spending. The implications of this study are critical for policymakers assessing the effectiveness of marketplace facilitator laws and economic nexus and their broader economic consequences. Limitations include potential lags in policy effects and unobserved differences in state enforcement intensity. Future research should explore long-term impacts and sector-specific responses to understand the full consequences of the *Wayfair* ruling on economic activity and public finance.

Table of Contents

I. Introduction

In 2018, the Supreme Court's *South Dakota v. Wayfair, Inc.* (*Wayfair*) decision overturned 51 years of state sales tax precedent. This precedent originated in *National Bellas Hess v. Department of Revenue of Illinois* (1967), where the Court established the physical presence rule, holding that states could not require businesses lacking a physical presence to collect and remit sales tax (Oyez, 2018). The physical presence rule was reaffirmed in 1992 in *Quill Corp. v. North Dakota* (*Quill*), where the Court determined that a state could not impose sales tax collection obligations on an out-of-state retailer unless it had a physical presence in the state (Oyez, 2018). However, in *Wayfair*, the Court eliminated the physical presence rule, holding that a business has a substantial nexus if it engages in significant economic activity within a state, allowing states to require remote sellers to collect and remit sales tax (Oyez, 2018). After *Wayfair*, states rapidly revised their sales tax codes by enacting marketplace facilitator and economic nexus laws to expand their taxing authority over remote sales. Marketplace facilitator laws require platforms like eBay and Amazon to collect and remit sales tax on behalf of third-party sellers (Fox et al., 2022). Economic nexus laws require remote sellers to collect and remit sales tax once they surpass specific sales thresholds within a state (Fox et al., 2022). In this post-*Wayfair* framework, a "substantial nexus" is established when a remote seller exceeds a monetary threshold specified within a state's tax code, such as surpassing $100,000 in sales or conducting more than 200 transactions annually, though the exact thresholds vary across states (Fox et al., 2022). These changes in state tax codes have profound economic implications for state tax revenues, business formation, and public investments. This study analyzes the national economic impact of these laws that followed *Wayfair* on state sales tax collection, entrepreneurship, and public expenditures related to social services, education, infrastructure, and security. The question,

"How have the marketplace facilitator and economic nexus laws enacted after *Wayfair* affected state sales tax revenues, entrepreneurship, and public expenditures on social services, education, infrastructure, and security across the 50 states?" guides this investigation.

Although the *Wayfair* decision has received significant academic attention, critical gaps remain in the literature. Since *Wayfair*, many researchers have focused on aggregate increases in sales tax revenues and the impact of new compliance regulations (Fox et al., 2022). However, many have neglected to study how these effects vary across states, how these legal changes affect entrepreneurship, and what changes have occurred in areas of critical public spending. Determining how marketplace facilitator laws and economic nexus laws have impacted sales tax revenue at the state level and their respective investments in crucial areas of public policy is essential for policymakers to decide on future tax policy and public spending changes. Further, assessing how *Wayfair* affects entrepreneurship provides insight into the challenges and opportunities created by the sales tax laws that ensued following the decision, including compliance burdens, market competition, and tax obligations. This research seeks to fill the current gaps in economic literature by examining state-specific variations, entrepreneurship, and public spending impacts by answering the guiding research question.

The investigation utilizes a two-way fixed effects difference-in-difference model, using the implementation of marketplace facilitator laws and economic nexus laws as the primary regressors to study these post-*Wayfair* policies on the outcome variables of state sales tax revenue, entrepreneurship, and public spending. The analysis uses data provided by the US Census Bureau relating to the outcome variables. Various control variables are implemented in the analysis, including state political party control, state gross domestic product, state personal income, state sales tax rate, and state economic nexus threshold. Overall, the research does not

find enough statistically significant evidence that would point to the implementation of marketplace facilitator laws and economic nexus laws being associated with changes in state sales tax revenues, entrepreneurship, or public spending in the areas of education, infrastructure, and security. The implementation of marketplace facilitator laws, however, are associated with a statistically significant decrease in per-capita spending in social services. The limitations of this study include the staggered implementation of these two post-*Wayfair* policies, as well as not accounting for enforcement intensity or compliance costs, which may influence the outcomes of this study. The remainder of this paper includes a review of the existing literature on the economic implications of *Quill* and *Wayfair* (Section II), a detailed section concerning the data sources (Sections III and IV), theoretical discussion (Section V), methodology (Section VI), results (Section VII), and the implications and limitations of the study (Section VIII).

II. Literature Review

a. The Impact of *Quill*

For over 50 years, the physical presence rule, established in *National Bellas Hess v. Department of Revenue of Illinois* and reaffirmed in *Quill*, governed sales tax collection and required that a seller have a physical presence in a state before being obligated to collect sales taxes (Oyez, 2018). Critics of *Quill* argued that the rule, though once appropriate, had become economically inefficient and inequitable in the face of the e-commerce revolution, which expanded at a compound annual growth rate of 11.9%, from $1.06 trillion in 2000 to $5.71 trillion in 2015 (Law Professors and Economists, South Dakota v. Wayfair, Inc., 2018; Fox, 2017). This rapid growth enabled tax avoidance by remote sellers, which distorted consumer behavior and weakened state revenue systems (Fox, 2017). With fewer compliance burdens,

online sellers benefited from tax advantages and gained market power over local retailers, which resulted in inefficient and uneven competition (Fox, 2017). Further, critics pointed out that because of the explosion of e-commerce sales, in part due to *Quill*, the foregone state and local sales tax was substantial, totaling an estimated $68.8 billion nationwide in 2015 (Fox, 2017; Bruce et al., 2009). This foregone revenue threatened state budgets, restricting state expenditures on public services like education and infrastructure that affect individuals daily (Fox, 2017).

As litigation for *Wayfair* progressed, legal scholars and economists urged the Supreme Court to overturn *Quill*'s outdated physical presence requirement (Law Professors and Economists, South Dakota v. Wayfair, Inc., 2018 Law professors and economists responded to the question before the Court, "Whether this Court should overrule the dormant Commerce Clause holding of *Quill Corp. v. North Dakota*" (Law Professors and Economists, South Dakota v. Wayfair, Inc., 2018). Thus, empirical evidence (Fox, 2017) and scholarly opinion (Law Professors and Economists, South Dakota v. Wayfair, Inc., 2018) converged to build a compelling case against *Quill*'s continued relevance.

b. The Effect of *Wayfair*

In 2018, the Supreme Court overruled the precedent of *Quill*, establishing a new jurisprudence regarding the dormant Commerce Clause and allowing states to force firms with a substantial economic nexus in the state to collect sales tax, regardless of an established physical presence (Oyez, 2018). Following the decision, states began implementing economic nexus and marketplace facilitator laws, with adoption occurring over the next two years (Fox et al., 2022). With this, Fox et al. (2022) find sales tax revenues increased nationally by 7.9%, with the most significant increases occurring in states with the strictest enforcement measures following the adoption of these laws. Further, they discover that shifting the collection requirement from

consumers to sellers improved compliance, a pattern consistent with prior tax compliance literature (Fox et al., 2022). Using barcode-level scanner data and observing that "aggregate pre-tax prices did not change when states imposed statutory incidence on sellers", they also find that the tax burden is fully passed through to consumers, with wealthier households paying more in absolute terms, a finding that makes the effect somewhat progressive and suggests that higher-income households had disproportionately benefited from the *Quill*-era rules (Fox et al., 2022).

Luchs-Nuñez (2021) explores the behavioral and financial responses of consumers, firms, and investors following *Wayfair*. Her analysis finds that consumers were generally unaffected by *Wayfair*. Yet, e-commerce sellers have not expanded into new states following *Wayfair* significantly. Still, those that did were 14% more likely to grow into states with high sales tax rates, potentially showing that these rates deterred expansion for some, but benefits outweighed the tax liability for others (Luchs-Nuñez, 2021). Regarding investments, Luchs-Nuñez (2021) finds that investors may have initially overreacted to *Wayfair* as the stock returns of e-commerce sellers had returns ranging from -1.0 to -2.6%, with such sellers that lacked locations across different states experiencing greater declines (Luchs-Nuñez, 2021). Extending the examination of market reactions, Kubick et al. (2024) investigate volatility patterns surrounding key *Wayfair*-related dates. This study confirms that of Luchs-Nuñez and identifies three particular dates surrounding *Wayfair* where markets experienced the greatest volatility. These dates include when the Supreme Court agreed to hear the case on January 12, 2018, the day of oral arguments on April 17, 2018, and the day of the final decision on June 21, 2018 (Kubick et al., 2024). The study also found increased forecasting errors and earnings estimates for firms most impacted by *Wayfair*, confirming the uncertainty seen by Luchs-Nuñez (Kubick et al., 2024).

While these studies largely highlight the benefits of *Wayfair*, other scholars have raised concerns about its unintended consequences. Critics generally agree that the Supreme Court's decision in *Wayfair* is substantially more appropriate given the current economic context compared to *Quill*, but concerns include compliance burdens on small and mid-size businesses, state tax complexities, and the potential for over-taxation (Ventulan, 2019). Although some believed *Wayfair* would primarily affect large firms, Inscore (2019), Kim (2020), and Butzler (2019) emphasize the disproportionate compliance burdens on small and mid-size businesses, who face complex tax rules across over 9,600 jurisdictions. Fox et al. (2022) provide a comprehensive analysis of *Wayfair*'s effects, yet they focus primarily on national revenue gains and shifts in compliance mandates; they do not analyze how states used these new revenues and their impact on public spending. Despite growing literature on national-level impacts, few studies have examined how post-*Wayfair* tax revenue changes vary across states or how these changes affect public expenditures and entrepreneurship. This study addresses that gap by analyzing how post-*Wayfair* tax revenue changes have influenced state-level spending in education, infrastructure, and public safety, as well as their potential effects on firm formation and business activity. This study tests the hypothesis that *Wayfair*-induced sales tax laws increased state tax revenues and public expenditures, while having mixed effects on entrepreneurship.

III. Data Preview

This investigation uses panel data from 2008 to 2023 to examine state sales tax revenues, entrepreneurship, and public spending. This section introduces the datasets, outcome variables, policy indicators, and control variables used to estimate the impact of post-*Wayfair* sales tax laws. All monetary values are later adjusted for inflation to ensure comparability across time

periods, and non-categorical variables are expressed in per-capita terms. Avalara (2025) and the Sales Tax Institute (2025) provide data on the timing and adoption of marketplace facilitator laws and economic nexus laws based on state tax codes and the annual state-level economic nexus thresholds. Marketplace facilitator laws and economic nexus laws are the two state legislative responses that followed *Wayfair*. Marketplace facilitator laws require companies like eBay and Amazon that facilitate third-party sales to collect and remit sales tax on behalf of the sellers who use their platforms (Fox et al., 2022). Economic nexus laws require out-of-state sellers to collect and remit sales tax if they exceed a specific monetary threshold in a state (i.e., substantial nexus), usually defined by a minimum number of transactions or a dollar amount of sales, or a combination of the two (Fox et al., 2022).

The investigation consists of two primary treatment indicators, one representing each type of law. $MarketFac_{st}$ indicates whether a state had implemented a marketplace facilitator law in a given year, and $EconNexus_{st}$ indicates whether a state had implemented an economic nexus law in a given year. Each variable equals 1 in years when the law is in effect and 0 otherwise. States are assigned to the treatment group if they implemented a marketplace facilitator law or an economic nexus law between 2018 and 2023. States that did not adopt either law during this period serve as the control group. These variables serve as key treatments to identify the impact of expanded state sales tax authority on sales tax revenues, entrepreneurship, and public spending. Avalara (2025) and the Sales Tax Institute (2025) datasets draw directly from state statutes to determine the exact year of adoption for each law across the states. However, states vary in their specific thresholds, exemptions, and enforcement aggressiveness, meaning the variables may not capture differences in the practical implementation of these laws, a limitation acknowledged in this analysis. While the policy indicator variables capture the adoption of

marketplace facilitator and economic nexus laws, they do not fully reflect variation in enforcement intensity or administrative compliance efforts across states, which may influence the observed outcomes.

This study examines state sales tax revenues, $TaxRev_{st}$, as a key outcome variable, representing state sales tax revenue in state *s* in year *t*. The US Census Bureau's *Annual Survey of State Government Tax Collections Data* (2024) provides the total state sales tax revenues in each of the 50 states from 1967 to 2023. The dataset consists of each state's tax revenue data by year, categorized by tax type, including sales tax (US Census Bureau, 2024). The revenues are listed in thousands of US dollars in the year reported, unadjusted for inflation. This data permits the measurement of $TaxRev_{st}$ in each state before and after *Wayfair* to examine the impact of marketplace facilitator laws and economic nexus laws on state sales tax revenues.

Entrepreneurship ($Entrepreneur_{st}$) also serves as a critical outcome variable in this study, representing firm establishments in state *s* in year *t*. The US Census Bureau's *BDS Explorer* (2022) provides the total number of firms established from 1978 to 2022, permitting the measurement of firm creation as a representation of entrepreneurship. Specifically, that data provides "Establishment Births" as the primary measure of entrepreneurial activity. This measure best captures the creation of new business locations, which aligns closely with the concept of entrepreneurship as the decision to enter the market. The database provides filtering by numerous firm size categories, among other characteristics (US Census Bureau, 2022). Given that startups are typically very small at inception, this study restricts the sample to firms with an Initial Firm Size of 1 to 4 employees. This fine-level categorization allows for a more precise focus on genuine entrepreneurial ventures, as opposed to larger new entities that may be expansions of existing businesses. $Entrepreneur_{st}$ is measured annually by state before and after *Wayfair*,

allowing for the analysis of the impact of marketplace facilitator laws and economic nexus laws on entrepreneurship.

Public spending outcomes constitute this study's third major category of variables. This investigation focuses on *Wayfair*'s impact on four critical aspects of public policy: social services, education, security, and infrastructure, denoted as $Social_{st}$, Edu_{st}, $Security_{st}$, and $Infrastructure_{st}$, respectively. These variables represent their respective spending type in state *s* in year *t*. Definitions for these categories follow the US Census Bureau's *Annual Survey of State and Local Government Finance* (2022). "Social services" includes public welfare spending (i.e., cash assistance and vendor payments), hospital expenditures, and spending on health, social insurance administration, and veterans' services. "Education" includes spending on higher education, elementary and secondary education, and libraries. "Infrastructure" consists of expenditures relating to highways, airports, parking, sea and inland port facilities, and transit subsidies. "Security" includes spending on police and fire protection, corrections (e.g., prisons), and protective inspection and regulation. The database also provides the total state expenditures for all 50 states from 1999 to 2022 (US Census Bureau, 2023). It categorizes annual expenditures by type in each state (US Census Bureau, 2023). The expenditures are listed in thousands of US dollars in the year reported, unadjusted for inflation. This data permits the measurement of state public spending outcomes and the examination of the impact of marketplace facilitator laws and economic nexus laws.

In order to maximize confidence in the analysis results, this study controls for several state-level characteristics that could independently influence the outcomes of interest in state *s* in year *t*. These include state political party control ($PartyControl_{st}$), state gross domestic product (GDP_{st}), state personal income ($PersInc_{st}$), state sales tax rate ($Rate_{st}$), and state thresholds for

economic nexus laws ($Threshold_{st}$). State population is also included as a control, as all variables except those that are categorical are expressed in per-capita terms. Ballotpedia's *Gubernatorial and Legislative Party Control of State Government* (2025) provides political party control data at the state level from 2008 to 2025. States with a Republican trifecta (i.e., a legislature and governorship of the same political party) were assigned a 1 in year *t* in the dataset, states with a Democrat trifecta were assigned a 2 in year *t*, and states with a legislature and governorship of different political parties were assigned a 0 in year *t*. Controlling for political party control accounts for differences in fiscal policy priorities that could affect tax collection, business formation, and public spending levels across states.

The Bureau of Economic Analysis's *State Annual Summary Statistics: Personal Income, GDP, Consumer Spending, Price Indexes, and Employment* (2025) provides gross domestic product (GDP) and personal income data by state from 1998 to 2024. Wealthier states may experience different tax revenue growth and entrepreneurial trends independent of policy changes that followed *Wayfair*, making these economic conditions essential to control for. Additionally, state sales tax rates ($Rate_{st}$) and economic nexus law thresholds ($Threshold_{st}$) were collected and used as control variables to account for differences in the pre-existing tax environment across states. These rates and thresholds were obtained from the Tax Foundation (2008-2023) and the Sales Tax Institute (2025) based on state sales tax codes and are included to control for differences in pre-existing tax policy structures across states and different enforcement structures implemented after *Wayfair*. States with higher baseline sales tax rates or lower nexus thresholds may have experienced different revenue or entrepreneurship responses following *Wayfair*.

The US Census Bureau's *State Intercensal Tables: 2000-2010* (2011), *State Population Totals and Components of Change: 2010-2019* (2019), and *State Population Totals and Components of Change: 2020-2024* (2024) provide state population estimates from 2000 to 2024 based on national census data and population trends. Population size affects both the tax base and the potential for entrepreneurial activity, making it essential to control for. All variables in state *s* in year *t* were divided by the corresponding state population in year *t* to express them in per-capita terms and control for differences in population, apart from the variables $PartyControl_{st}$, $Rate_{st}$, and $Threshold_{st}$. $Entrepreneur_{st}$ is expressed per 10,000 people (e.g., a value of 13 indicates 13 new firms per 10,000 residents). Expressing outcomes in per-capita terms is essential for comparability, as states differ widely in population size. Without scaling population, differences in tax revenues, firm creation rates, or public spending could reflect state size differences rather than actual policy impact differences. To ensure comparability across time, tax revenue, personal income, and public expenditure data are adjusted for inflation to January 2025 US dollars. State GDP data are converted to real GDP using the implicit price deflator, set at Q4 2024 price levels. The Federal Reserve Bank of St. Louis' Federal Reserve Economic Data provides GDP deflator trends in their *Gross Domestic Product: Implicit Price Deflator* (2025) and consumer price index (CPI) trends in their *Table Data - Consumer Price Index for All Urban Consumers: All Items in US City Average* (2025).

IV. Data Analysis

Of the 50 US states, 45 collect state sales taxes, while five (Alaska, Delaware, Montana, New Hampshire, and Oregon) do not. These five non-collecting states are the control group, since they were unaffected by marketplace facilitator and economic nexus laws following the *Wayfair* decision. The remaining 45 states form the treatment group. Table 1 (see Appendix A)

presents summary statistics for the outcome and control variables across pre-treatment and post-treatment periods from 2008 to 2023. Across all states, mean per-capita sales tax revenue increased from $1,082.00 in the pre-treatment years to $1,218.15 in the post-treatment period, showing a potential rise in overall collections. Similarly, the average number of firm establishments with 1 to 4 employees (per 10,000 people) increased from 13.56 to 14.66.

When separating by treatment status, treated states show similar upward trends. Sales tax revenue rose from $1,202.23 to $1,353.50 and firm establishments from 13.23 to 14.23. Control states, which do not collect sales tax and therefore do not report revenue figures, also experienced an increase in firm establishments from 16.58 to 18.51 Control states had higher pre-treatment means for several variables, such as public spending and personal income, showing significant baseline differences that are accounted for in the regression analysis. For spending outcomes, treated states saw growth in social services spending (from $2,699.55 to $3,299.82), education (from $1,370.54 to $1,396.37), and GDP, while infrastructure and security spending remained relatively flat or declined slightly. Economic nexus thresholds averaged $128,962.96 across treated states in the post-treatment period. These patterns suggest that, although treated and control states differ in baseline levels, the directional trends in outcomes, particularly in firm formation and public spending, are generally consistent, lending preliminary support to the parallel trends assumption required for difference-in-differences analysis.

To validate the use of a two-way fixed effects (TWFE) difference-in-differences (DID), parallel trends tests are conducted across five outcome variables: firm establishments, social services spending, education spending, infrastructure spending, and security spending (see Appendix B for the results in Table 2). Each regression includes a third-degree polynomial in relative time and interactions between time terms and treatment status. These interaction terms

identify whether treated and control states followed similar trajectories before implementing marketplace facilitator and economic nexus laws, with statistically significant interaction terms indicating a failure to satisfy the parallel trends assumption. All three interaction terms are statistically insignificant for firm establishments (1 to 4 employees), social services, education, and security spending. These results indicate that entrepreneurship trends in treated and control states moved in parallel before treatment, supporting the validity of DID estimation for this outcome. The interaction term for RT3 and Treatment is statistically significant at the 10% level, providing weak statistical evidence of potential divergence in pre-treatment trends. Although this does not constitute a strong violation of the assumption of parallel trends, it does warrant caution when interpreting treatment effects for infrastructure spending.

V. Theoretical Discussion

This study examines the implications of Wayfair in the context of optimal taxation theory and the broader objective of policymakers to maximize social welfare. Under optimal taxation theory, an effective tax system should raise sufficient revenue while minimizing economic distortions and maintaining equity. Before Wayfair, the physical presence rule exempted many online retailers from collecting and remitting sales taxes, giving them a competitive advantage over brick-and-mortar businesses that faced collection mandates (Fox, 2017). As discussed earlier, this created market inefficiencies by allowing tax avoidance to influence consumer behavior (Fox, 2017). *Wayfair* aimed to correct this distortion by allowing states to collect sales tax based on the substantial economic nexus rule rather than physical presence (Fox et al., 2022). This shift was intended to restore competitive neutrality and reduce the revenue losses that occurred under *Quill* (Fox et al., 2022). In theory, these changes could influence firm behavior,

particularly among small businesses, by altering the cost structure and compliance environment, as emphasized in the literature.

Marketplace facilitator and economic nexus laws implemented after *Wayfair* may affect entrepreneurship. Increased compliance requirements could raise fixed costs and discourage market entry, especially for small firms. However, with *Wayfair*'s equalizing effect on tax obligations for all sellers, these laws could reduce competition distortions and improve market conditions, especially for larger firms. The fiscal policy implications are also substantial. An effective remote sales tax collection expansion should boost general fund revenues, enabling greater public investment in social services, education, infrastructure, and security. Such spending could produce social welfare gains if these funds are allocated effectively.

However, the effects of *Wayfair* are unlikely to be uniform across states. Economic structure, tax policy, and administrative capacity variation suggest heterogeneous impacts. States without a personal income tax, and therefore more reliant on sales tax, are expected to benefit disproportionately from expanded taxing authority. Likewise, states that previously lacked access to major online sellers will likely experience larger post-*Wayfair* revenue gains. This study tests whether *Wayfair*-induced tax reforms increased state sales tax revenues, affected small business formation, and expanded public spending in treated states relative to control states. This study expects an increase in state sales tax revenue following *Wayfair*, consistent with any effective tax policy. Entrepreneurship could decrease with increased compliance burdens, yet it may increase due to the level of playing field offered under *Wayfair*. The magnitude of each effect should determine the results. Finally, the study expects increased public expenditures following increased sales tax collections.

VI. Methodology

This study tests the hypothesis that *Wayfair*-induced sales tax laws increased state tax revenues and public expenditures, while having mixed effects on entrepreneurship. To do this, it employs a two-way fixed effects (TWFE) difference-in-difference (DID) analysis to determine *Wayfair*'s impact on state sales tax revenues, entrepreneurship, and public expenditures relating to social services, education, security, and infrastructure. The investigation's linear and log models are shown in Equations 1 and 2, respectively.

$$Y_{st} = B_0 + B_1 MarketFac_{st} + B_2 EconNexus_{st} + State_s + Year_t + X_{st} + Error_{st} \quad (1)$$

$$\text{Log}(Y_{st}) = B_0 + B_1 MarketFac_{st} + B_2 EconNexus_{st} + State_s + Year_t + X_{st} + Error_{st} \quad (2)$$

Y_{st} represents the outcome variables $TaxRev_{st}$, $Entrepreneur_{st}$, $Social_{st}$, Edu_{st}, $Security_{st}$, and $Infrastructure_{st}$. $MarketFac_{st}$ and $EconNexus_{st}$ indicate whether a state s has implemented a marketplace facilitator and economic nexus law in a given year t. Each variable equals 1 in years when the law is in effect and 0 otherwise. $State_s$ and $Year_t$ are the fixed effects for state and year. X_{st} represents the control variables state political party control ($PartyControl_{st}$), state gross domestic product (GDP_{st}), state personal income ($PersInc_{st}$), state sales tax rate ($Rate_{st}$), and state thresholds for economic nexus laws ($Threshold_{st}$).

In the linear model (Equation 1), the coefficients represent the estimated absolute change in the outcome variable associated with each independent variable, holding other factors constant. For example, a coefficient of 100 on $MarketFac_{st}$ would imply that the law is associated with a $100 increase in per-capita sales tax revenue.

In the log model (Equation 2), the coefficients approximate percentage changes in the outcome. For example, a coefficient of 0.05 on $MarketFac_{st}$ would indicate that the law is associated with a 5% increase in the dependent variable.

VII. Results

This section presents the results of the TWFE DID analysis that examines the effect of the marketplace facilitator and economic nexus laws that followed *Wayfair* on six outcome variables: state sales tax revenues, entrepreneurship, and public expenditures related to social services, education, infrastructure, and security. Table 3 (see Appendix C) displays results from the linear model, while Table 4 (see Appendix D) presents the log-transformed results. Both models control for state and year fixed effects and include state-level controls for GDP, personal income, political party control, state sales tax rate, and economic nexus threshold.

The Marketplace Facilitator Law and Economic Nexus Law variables are statistically insignificant for state sales tax revenue in both the linear and log models. This suggests that these policies did not produce immediate or uniform increases in state sales tax revenue across the treated states. However, the state sales tax rate and GDP are positively and significantly associated with revenue in both models. These results suggest that while the structure of sales tax enforcement shifted after *Wayfair*, the state sales tax rate and broader economic conditions may remain the primary drivers of state tax collections.

Across both models, the treatment variables are not statistically significant for Entrepreneurship, indicating no apparent effect of marketplace facilitator or economic nexus laws on firm establishments for firms with 1 to 4 employees. However, the control variables of personal income and state sales tax rate significantly affect both models. This is consistent with

expectations that higher tax burdens may discourage small business formation. In addition, the effect of personal income may mean that higher income levels support greater business activity.

The Marketplace Facilitator Law is associated with a statistically significant reduction in per-capita social services spending in both models. In the linear model, the estimated effect is a \$297.70 decrease ($p < 0.05$), while in the log model, the law is associated with an 8.25% reduction ($p < 0.05$). This consistent negative association may reflect delayed budget adjustments, reallocation of funds, or differences in implementation capacity across states. The Economic Nexus Law is not significant in either model. Other controls, such as GDP and personal income, are positively related to spending, though only marginally significant.

Treatment variables again show no statistically significant effect on education spending. Personal income is positively associated with education spending in both models, while the state sales tax rate is negatively associated with education spending, which may suggest shifting priorities in state budgets.

The treatment variables are statistically insignificant in all models for infrastructure and security. However, in the infrastructure model, GDP is positively associated with spending in the linear model, and party control is negatively associated in the log model. In the security model, personal income is negatively associated with spending in both models, though the mechanism remains unclear and may reflect budget substitution effects.

All models show strong explanatory power, with adjusted R-squared values ranging from 0.876 to 0.962. Each regression passes the joint F-test for overall model significance at the 1% level, indicating that the included variables explain substantial variation in the outcomes.

Overall, the results suggest that the tax policies that followed *Wayfair* had limited direct effects on sales tax revenues and entrepreneurship. However, marketplace facilitator laws are associated with declining social services spending. In total, the state sales tax rate, GDP, and personal income consistently influence revenue and expenditure outcomes, highlighting the importance of the broader economic context. While the treatment effects are generally modest or statistically insignificant, the findings point to heterogeneous state-level responses and the potential for lagged or indirect effects not fully captured in the immediate post-*Wayfair* period.

VIII. Conclusion and Implications

This study evaluates the impact of *Wayfair*-related marketplace facilitator and economic nexus laws on state tax revenues, entrepreneurship, and public expenditures using a two-way fixed effects difference-in-differences (TWFE DID) framework. The guiding research question, "How have the marketplace facilitator and economic nexus laws enacted after *Wayfair* affected state sales tax revenues, entrepreneurship, and public expenditures on social services, education, infrastructure, and security across the 50 states?" guides the research process with the analysis testing the ultimate hypothesis. By analyzing outcomes from 2008 to 2023 across treated and control states, the study isolates the effects of these *Wayfair* policies while accounting for state-specific and time characteristics and economic conditions.

The results show limited direct effects of the marketplace facilitator and economic nexus laws on state sales tax revenue and entrepreneurship, suggesting that implementing these laws did not produce immediate changes in these areas. However, the marketplace facilitator law is consistently associated with a significant decline in social services spending, which may reflect transitional budget constraints or shifts in state fiscal priorities. Across all models, state sales tax

rates, GDP, and personal income emerged as consistent revenue and spending outcomes predictors. While the treatment effects were generally modest, the high explanatory power of the models suggests that the included economic controls capture much of the variation in outcomes. These findings have several implications for tax policy. While *Wayfair* expanded state tax authority, it did not automatically result in significant revenue gains or entrepreneurial effects.

The limitations of this study are essential to acknowledge. The effects of *Wayfair*-related laws may vary over time and across states, and some impacts may emerge gradually or in sectors not captured by the selected outcome variables. In addition, while this analysis accounts for significant economic covariates, unobserved factors, such as enforcement intensity or compliance costs, could still influence the outcomes. Future research should analyze the long-term effects of these policies as more time and data become available. Firm behavioral responses and remote sales taxation distribution should also be looked into. This study suggests that the consequences of *Wayfair* are more subtle than initially expected. While the ruling resolved major economic problems related to the physical presence rule under *Quill*, its effects seem to depend on broader state economic conditions. These results contribute to a better understanding of modern tax enforcement and its interaction with financial outcomes.

References

Avalara (2025, January 31). *Your guide to state laws and regulations for marketplace facilitators and marketplace sellers*. https://www.avalara.com/us/en/learn/guides/state-by-state-guide-to-marketplace-facilitator-laws.html

Ballotpedia (2025, January 1). *Gubernatorial and legislative party control of state government*. https://ballotpedia.org/Gubernatorial_and_legislative_party_control_of_state_government#State_government_trifectas,_pre-2010

Bureau of Economic Analysis (2025, March 8). *State Annual Summary Statistics: Personal Income, GDP, Consumer Spending, Price Indexes, and Employment.* https://apps.bea.gov/itable/

Brief of Amici Curiae Law Professors and Economists in Support of Petitioner. *South Dakota v. Wayfair, Inc.* Supreme Court of the United States (2018). https://www.supremecourt.gov/DocketPDF/17/17-494/37603/20180305141434827_Brief%20of%20Amici%20Curiae%20Law%20Professors%20and%20Economists%20iso%20Petitioner.PDF

Bruce, D., Fox, W. F., & Luna, L. (2009, April 13). State and Local Government Sales Tax Revenue Losses from Electronic Commerce. Haslam College of Business. https://haslam.utk.edu/wp-content/uploads/2019/05/20090401.pdf

Butzler, A. S. (2019). The Eradication of Online Retailers' Tax Shelter: How South Dakota V. Wayfair Eliminated the Physical Presence Standard and Reinterpreted the Commerce Clause to Allow Collection of State Sales Tax on Remote Sellers. *Gonzaga Law Review*, *54*(2).

Fox, W. F. (2017, November 6). Inability to Collect Sales Tax on Remote Sales Still Harms the Economy. *State Tax Notes*, 575-582.

Fox, W. F., Hargadena, E. P., & Luna, L. (2022). Statutory incidence and sales tax compliance: Evidence from Wayfair. *Journal of Public Economics*, *213*. https://doi.org/10.1016/j.jpubeco.2022.104716

Federal Reserve Bank of St. Louis (2025, January). Gross Domestic Product: Implicit Price Deflator. Federal Reserve Economic Data. https://fred.stlouisfed.org/series/GDPDEF

Federal Reserve Bank of St. Louis (2025, January). Gross Domestic Product: Implicit Price Deflator. Federal Reserve Economic Data. https://fred.stlouisfed.org/series/GDPDEF

Federal Reserve Bank of St. Louis (2025, February). Table Data - Consumer Price Index for All Urban Consumers: All Items in US City Average. Federal Reserve Economic Data. https://fred.stlouisfed.org/data/cpiaucns

Inscore, J. (2019). The Amazon Argument: An Examination of South Dakota v. Wayfair and a Discussion of its Implications. *Campbell Law Review*, *41*(2), 531.

Kim, K. (2020). The Disproportionate Sales Tax Compliance Burden on Small- and Medium-Sized Online Businesses After Wayfair. *Science and Technology Law Review*, *22*(1), 181–230. https://doi.org/10.52214/stlr.v22i1.8057

Kubick, T. R., Omer, T. C., & Yazzie, C. (2021). Economic Consequences of Expanding Sales Tax Nexus: Evidence from Stock Price Reactions to the Wayfair Decision. *Journal of Accounting and Public Policy, Forthcoming*. https://doi.org/http://dx.doi.org/10.2139/ssrn.3800440

Luchs-Nuñez, J. (2024). Economic Effects of Sales Tax Collection Requirements: Evidence from Wayfair. *University of Connecticut School of Business Research Paper No. 21-17*. https://doi.org/http://dx.doi.org/10.2139/ssrn.3924416

Oyez (2018). South Dakota v. Wayfair, Inc. https://www.oyez.org/cases/2017/17-494

Sales Tax Institute (2025, April 25). *Economic Nexus State by State Chart*. https://www.salestaxinstitute.com/resources/economic-nexus-state-guide

Tax Foundation (2008). State and Local General Sales Tax Collections Per Capita Fiscal Year 2008. https://files.taxfoundation.org/legacy/docs/s%26l_salestax_coll_pc_fy2008-20110301.pdf

Tax Foundation (2009, October 16). Updated State and Local Option Sales Tax. https://taxfoundation.org/research/all/state/updated-state-and-local-option-sales-tax/

Tax Foundation (2010, March 29). State General Sales and Use Tax Rates. https://taxfoundation.org/data/all/state/state-general-sales-and-use-tax-rates/

Tax Foundation (2013, March 14). State and Local Sales Tax Rates, 2011 – 2013. https://taxfoundation.org/data/all/state/state-and-local-sales-tax-rates-2011-2013/

Tax Foundation (2014, March 18). State and Local Sales Tax Rates, 2014. https://taxfoundation.org/data/all/state/state-and-local-sales-tax-rates-2014/

Tax Foundation (2015, April 8). State and Local Sales Tax Rates, 2015. https://taxfoundation.org/data/all/state/state-and-local-sales-tax-rates-2015/

Tax Foundation (2016, March 9). State and Local Sales Tax Rates, 2016. https://taxfoundation.org/data/all/state/state-and-local-sales-tax-rates-2016

Tax Foundation (2017, January 31). State and Local Sales Tax Rates, 2017. https://taxfoundation.org/data/all/state/state-and-local-sales-tax-rates-in-2017/

Tax Foundation (2018, February 13). State and Local Sales Tax Rates, 2018. https://taxfoundation.org/data/all/state/state-and-local-sales-tax-rates-2018/

Tax Foundation (2019, January 30). State and Local Sales Tax Rates, 2019. https://taxfoundation.org/data/all/state/sales-tax-rates-2019/

Tax Foundation (2020, January 15). State and Local Sales Tax Rates, 2020. https://taxfoundation.org/data/all/state/2020-sales-taxes/

Tax Foundation (2021, January 6). State and Local Sales Tax Rates, 2021. https://taxfoundation.org/data/all/state/2021-sales-taxes

Tax Foundation (2022, February 3). State and Local Sales Tax Rates, 2022. https://taxfoundation.org/data/all/state/2022-sales-taxes/

Tax Foundation (2023, February 7). State and Local Sales Tax Rates, 2023. https://taxfoundation.org/data/all/state/2023-sales-taxes/

US Census Bureau (2022). *Business Dynamics Statistics*. BDS Explorer. https://bds.explorer.ces.census.gov/?year=2022&xaxis-id=sector&xaxis-selected=31-33,44-45,54,72&group-id=fage&group-selected=010,065,070,075,150&group-group=2&measure-id=job_creation&chart-type=bar

US Census Bureau (2023). *Annual Survey of State and Local Government Finance Data*. https://www.census.gov/programs-surveys/gov-finances/data/datasets.html

US Census Bureau (2024, May 4). *Annual Survey of State Government Tax Collections Data*. https://www.census.gov/programs-surveys/stc/data/datasets.html?text-list-4071d3d4f8%3Atab=all#text-list-4071d3d4f8

US Census Bureau, Population Division (2011, September). State Intercensal Tables: 2000-2010. https://www.census.gov/data/tables/time-series/demo/popest/intercensal-2000-2010-state.html

US Census Bureau, Population Division (2019, December). State Population Totals and Components of Change: 2010-2019. https://www.census.gov/data/tables/time-series/demo/popest/2010s-state-total.html

US Census Bureau, Population Division (2024, December). State Population Totals and Components of Change: 2020-2024. https://www.census.gov/data/tables/time-series/demo/popest/2020s-state-total.html

Ventulan, J. (2019). South Dakota v. Wayfair: Way Unfair? *Southeastern University Law Review*, *46*, 235. https://doi.org/http://dx.doi.org/10.2139/ssrn.3334585

Appendix A: Table 1

Table 1: Summary Statistics of Outcome and Control Variables (2008-2023)

ALL STATES	Pre-Treatment Years		Post-Treatment Years	
	Mean	Std. Dev	Mean	Std. Dev
Sales Tax Revenue[1]	$1,082.00	$546.34	$1,218.15	620.36
Firm Establishments (1-4 EE)[2]	13.56	3.56	14.66	4.35
Social Services Spending[3]	$2,736.83	$729.20	$3,368.52	$933.30
Education Spending[3]	$1,411.10	$492.88	$1,435.21	$468.88
Infrastructure Spending[3]	$599.85	$406.83	$581.72	$345.63
Security Spending[3]	$321.59	$127.27	$315.80	$137.12
Personal Income[4]	$61,746.91	$7,495.98	$77,198.40	$11,683.80
GDP[4]	$70,474.32	$11,527.35	$72,693.82	$7,761.65
Sales Tax Rate[5]	5.06%	1.97%	5.09%	1.96%
Economic Nexus Threshold[6]	N/A	N/A	$116,066.67	$103,270.25

CONTROL STATES	Pre-Treatment Years		Post-Treatment Years	
	Mean	Std. Dev	Mean	Std. Dev
Sales Tax Revenue[1]	N/A	N/A	N/A	N/A
Firm Establishments (1-4 EE)[2]	16.58	3.43	18.51	3.85
Social Services Spending[3]	$3,072.43	$959.99	$3,986.88	$1,077.53
Education Spending[3]	$1,776.17	$672.88	$1,728.87	$593.69
Infrastructure Spending[3]	$1,084.81	$903.11	$951.26	$763.57
Security Spending[3]	$502.15	$233.63	$531.12	$266.12
Personal Income[4]	$62,690.32	$6,195.63	$73,160.18	$5,420.18
GDP[4]	$75,285.37	$16,044.02	$78,635.29	$11,273.06
Sales Tax Rate[5]	N/A	N/A	N/A	N/A
Economic Nexus Threshold[6]	N/A	N/A	N/A	N/A

TREATED STATES	Pre-Treatment Years		Post-Treatment Years	
	Mean	Std. Dev	Mean	Std. Dev
Sales Tax Revenue[1]	$1,202.23	$432.28	$1,353.50	$493.85
Firm Establishments (1-4 EE)[2]	13.23	3.42	14.23	4.19
Social Services Spending[3]	$2,699.55	$690.24	$3,299.82	$892.44
Education Spending[3]	$1,370.54	$451.76	$1,396.37	$451.40
Infrastructure Spending[3]	$545.97	$256.59	$540.66	$231.06
Security Spending[3]	$301.53	$89.51	$291.88	$86.96
Personal Income[4]	$61,642.09	$7,625.46	$72,642.00	$7,985.46
GDP[4]	$69,939.76	$10,803.78	$77,038.75	$11,737.94
Sales Tax Rate[5]	5.62%	1.08%	5.65%	1.04%
Economic Nexus Threshold[6]	N/A	N/A	$128,962.96	$100,919.74

Notes: All variables are expressed in per-capita terms except for Sales Tax Rate and Economic Nexus Threshold. Firm Establishments (1-4 EE) is per 10,000 people. All monetary variables are adjusted for inflation (USD 2025) and GDP data are expressed as real GDP set at Q4 2024 price levels. Alaska, Delaware, New Hampshire, Montanna, and Oregon are the control states, with the other 45 states comprising the treated staes.

Sources: 1. Annual Survey of State Government Tax Collections Data (2024) 2. BDS Explorer (2022) 3. Annual Survey of State and Local Government Finance (2023) 4. State Annual Summary Statistics: Personal Income, GDP, Consumer Spending, Price Indexes, and Employment (2025) 5. State and Local Sales Tax Rates (2008-2023) 6. Economic Nexus State by State Chart (2025)

Appendix B: Table 2

Table 2: Parallel Trends Tests

Regressors	Firm Establishments (1-4EE)	Social Services Spending	Education Spending	Infrastructure Spending	Security Spending
Treatment	-31,410.28	-$1,073.54	$336.02	$938.44	$156.19
	27,834.62	$189.65	$189.65	$122.65	$28.97
RT	-9,030.63	$550.22	-$36.04	$99.11	-$17.56
	20,865.72	$117.12	$117.12	$75.46	$18.14
RT^2	-2,288.02	$83.35	-$10.40	$24.90	-$3.99
	4,409.07	$22.18	$22.18	$13.93	$3.88
RT^3	-159.18	$4.60	-$0.54	$1.46	-$0.23
	269.34	$1.28	$1.28	$0.77	$0.24
RT*Treatment	2,089.54	-$331.53	$104.86	-$95.24	$15.67
	22,151.91	$119.42	$119.42	$78.22	$18.96
RT^2*Treatment	583.85	-$58.27	$21.77	-$23.91	$3.48
	4,669.19	$22.67	$22.67	$14.58	$4.05
RT^3*Treatment	43.32	-$3.38	$1.15	-$1.38*	$0.16
	284.81	$1.31	$1.31	$0.82	$0.25
Intercept	156,087.60	$4,451.06	$1,439.47	$447.43	$351.39
	26,005.65	$178.19	$178.19	$113.22	$24.16

*Notes: Robust standard errors are in parentheses. *, **, and *** indicate 10%, 5%, and 1% significance levels, respectively.*

Appendix C: Table 3

Table 3: Linear Results for All Outcome Variables

Regressors	Sales Tax Revenues	Entrepreneurship	Social Services Spending	Education Spending	Infrastructure Spending	Security Spending
Marketplace Facilitator Law	38.7567	-0.396	-297.6962**	0.5206	47.1684	-6.3561
	(47.7599)	(0.3147)	(124.3385)	(46.9686)	(44.4628)	(7.6904)
Economic Nexus Law	26.2078	-0.1322	134.6367	37.3613	-7.2658	-4.9465
	(55.0957)	(0.3103)	(109.4396)	(41.9237)	(38.1939)	(6.9617)
GDP	0.0179***	0	0.0189*	0.0011	0.0091**	0.0013
	(0.0055)	(0.0000)	(0.0101)	(0.0035)	(0.0043)	(0.0011)
Party Control	1.1069	-0.05	7.5864	0.503	-7.4212	-1.041
	(12.3825)	(0.0536)	(29.7209)	(14.1110)	(7.2055)	(2.2501)
Personal Income	-0.0022	0.0001**	0.0093	0.0120**	0.0012	-0.0024**
	(0.0097)	(0.0001)	(0.0115)	(0.0053)	(0.0041)	(0.0011)
State Sales Tax Rate	9,302.9221**	-25.1450*	-5114.8748	-6,950.2805**	214.6803	-348.5247
	(3,955.4829)	(13.5428)	(8,517.9675)	(3,030.4513)	(2,406.5413)	(458.1454)
Economic Nexus Threshold	0	0	0	0.0001	0.0001	0
	(0.0002)	(0.0000)	(0.0006)	(0.0002)	(0.0001)	(0.0000)
Intercept	-543.3204	5.3818**	675.8242	1,254.5607***	-145.509	310.4379***
	(472.9722)	(2.4730)	(698.7965)	(403.9301)	(351.2461)	(50.9830)
Adjusted R-Square	0.9327	0.9482	0.8763	0.9347	0.9374	0.9541
Overall Significance	4,656.82***	13,251.94***	928.92***	108.29***	7,331.77***	1,207.58***

Notes: Robust standard errors are in parentheses. *, **, and *** indicate 10%, 5%, and 1% significance levels, respectively.
Controls for state and year fixed effects are employed

Appendix D: Table 4

Table 4: Log Results for All Outcome Variables

Regressors	Sales Tax Revenues	Entrepreneurship	Social Services Spending	Education Spending	Infrastructure Spending	Security Spending
Marketplace Facilitator Law	0.0019	-0.0226	-0.0825**	-0.0101	0.069	-0.0052
	(0.0259)	(0.0202)	(0.0355)	(0.0266)	(0.0617)	(0.0237)
Economic Nexus Law	0.0245	-0.0077	0.0392	0.029	-0.0448	-0.0045
	(0.0311)	(0.0186)	(0.0331)	(0.0259)	(0.0587)	(0.0233)
GDP	0.0000***	0	0.0000**	0	0	0
	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
Party Control	-0.0024	-0.004	0.0036	0.0026	-0.0251*	-0.0048
	(0.0072)	(0.0037)	(0.0085)	(0.0073)	(0.0137)	(0.0065)
Personal Income	0	0.0000**	0	0.0000*	0	-0.0000**
	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
State Sales Tax Rate	8.1650***	-1.9189**	-1.9832	-5.4059***	0.1211	-1.4062
	(2.9377)	(0.8243)	(2.9042)	(2.0003)	(3.9718)	(1.3150)
Economic Nexus Threshold	0	0	0	0	0	0
	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
Intercept	5.4635***	2.1143***	7.0138***	7.2661***	5.7544***	5.6241***
	(0.2310)	(0.1467)	(0.2116)	(0.2005)	(0.3476)	(0.1454)
Adjusted R-Square	0.9442	0.9618	0.8941	0.9298	0.8975	0.9442
Overall Significance	410.74***	71,579.33***	15,903.04***	179.12***	4,378.69***	489.16***

Notes: Robust standard errors are in parentheses. *, **, and *** indicate 10%, 5%, and 1% significance levels, respectively.
Controls for state and year fixed effects are employed

Appendix E: SAS Codes

```
/*/////TaxRev/////*/


/*First Data*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.TaxRev
	dbms=xlsx
	replace;
sheet="TaxRev";
getnames=yes;
run;

Proc Sort Data=TaxRev;
	by State;
run;

Proc Transpose Data=TaxRev Out=TaxRev2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;

Data TaxRev3;
	Set TaxRev2;
	Year=input(_Name_,4.);
	TaxRev=Col1;
```

```
        keep State Year TaxRev;
Run;

/*Second Data*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Implemented
        dbms=xlsx
        replace;
sheet="Implemented";
getnames=yes;
run;

Proc Sort Data=Implemented;
        by State;
run;

/*Combine*/
Data Combine;
        Merge TaxRev3 Implemented;
        By State;

        /*MarketFac DID*/
        If MarketFacImplemented="-"  then MarketFac = 0;
        Else If Year>=MarketFacImplemented then MarketFac = 1;
        Else MarketFac=0;

        /*EconNexus DID*/
```

```
        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;

        Keep State Year TaxRev MarketFac EconNexus;
Run;

/*Model 1 - TaxRev w/o Control Variables*/
ods output ParameterEstimates=PEforModel1 DataSummary=ObsModel1
                FitStatistics=AdjRsqModel1 Effects=OverallSigModel1;
proc surveyreg data=Combine plots=none;
  class State Year / ref=first;
  cluster State;
  model TaxRev = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Model 2 - LogTaxRev w/o Control Variables*/
data LogCombine;
        set Combine;
        LogTaxRev = Log(TaxRev);
Run;

ods output ParameterEstimates=PEforModel2 DataSummary=ObsModel2
                FitStatistics=AdjRsqModel2 Effects=OverallSigModel2;
proc surveyreg data=LogCombine plots=none;
  class State Year / ref=first;
```

```
  cluster State;
  model LogTaxRev = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Control Variables Uploading*/

/*Third Data - Party Control*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.PartyControl
        dbms=xlsx
        replace;
sheet="PartyControl";
getnames=yes;
run;

Proc Sort Data=PartyControl;
        by State;
run;

Proc Transpose Data=PartyControl Out=PartyControl2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;
```

```
Data PartyControl3;
	Set PartyControl2;
	Year=input(_Name_,4.);
	PartyControl=Col1;
	keep State Year PartyControl;
Run;

/*Fourth Data - GDP*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.GDP
	dbms=xlsx
	replace;
sheet="GDP";
getnames=yes;
run;

Proc Sort Data=GDP;
	by State;
run;

Proc Transpose Data=GDP Out=GDP2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;
```

```
Data GDP3;
	Set GDP2;
	Year=input(_Name_,4.);
	GDP=Col1;
	keep State Year GDP;
Run;

/*Fifth Data - Personal Income*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.PersonalIncome
	dbms=xlsx
	replace;
sheet="PersonalIncome";
getnames=yes;
run;

Proc Sort Data=PersonalIncome;
	by State;
run;

Proc Transpose Data=PersonalIncome Out=PersonalIncome2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;
```

```
Data PersonalIncome3;
	Set PersonalIncome2;
	Year=input(_Name_,4.);
	PersonalIncome=Col1;
	keep State Year PersonalIncome;
Run;

/*Sixth Data - Population*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Pop
	dbms=xlsx
	replace;
sheet="Pop";
getnames=yes;
run;

Proc Sort Data=Pop;
	by State;
run;

Proc Transpose Data=Pop Out=Pop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;
```

```
Data Pop3;
	Set Pop2;
	Year=input(_Name_,4.);
	Pop=Col1;
	keep State Year Pop;
Run;

/*Seventh Data - TaxRev/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.TaxRevPop
	dbms=xlsx
	replace;
sheet="TaxRevPop";
getnames=yes;
run;

Proc Sort Data=TaxRevPop;
	by State;
run;

Proc Transpose Data=TaxRevPop Out=TaxRevPop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;
```

```
Data TaxRevPop3;
	Set TaxRevPop2;
	Year=input(_Name_,4.);
	TaxRevPop=Col1;
	keep State Year TaxRevPop;
Run;

/*Eighth Data - GDP/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.GDPPop
	dbms=xlsx
	replace;
sheet="GDPPop";
getnames=yes;
run;

Proc Sort Data=GDPPop;
	by State;
run;

Proc Transpose Data=GDPPop Out=GDPPop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;
```

```
Data GDPPop3;
	Set GDPPop2;
	Year=input(_Name_,4.);
	GDPPop=Col1;
	keep State Year GDPPop;
Run;

/*Ninth Data - Personal Income/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.PersonalIncomePop
	dbms=xlsx
	replace;
sheet="PersIncPop";
getnames=yes;
run;

Proc Sort Data=PersonalIncomePop;
	by State;
run;

Proc Transpose Data=PersonalIncomePop Out=PersonalIncomePop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;
```

```
Data PersonalIncomePop3;
	Set PersonalIncomePop2;
	Year=input(_Name_,4.);
	PersonalIncomePop=Col1;
	keep State Year PersonalIncomePop;
Run;

/*Tenth Data - Tax Rates*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Rate
	dbms=xlsx
	replace;
sheet="Rate";
getnames=yes;
run;

Proc Sort Data=Rate;
	by State;
run;

Proc Transpose Data=Rate Out=Rate2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;
```

```
Data Rate3;
	Set Rate2;
	Year=input(_Name_,4.);
	Rate=Col1;
	keep State Year Rate;
Run;

/*Eleventh Data - EN Threshold*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Threshold
	dbms=xlsx
	replace;
sheet="Threshold";
getnames=yes;
run;

Proc Sort Data=Threshold;
	by State;
run;

Proc Transpose Data=Threshold Out=Threshold2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;
```

```
Data Threshold3;
        Set Threshold2;
        Year=input(_Name_,4.);
        Threshold=Col1;
        keep State Year Threshold;
Run;

/*NEW MODELS*/

/*Combining Everything*/

/* Step 1: Merge TaxRev3 with Implemented (already done) */
Data Combine;
        Merge TaxRev3 Implemented;
        By State;

        If MarketFacImplemented="-"  then MarketFac = 0;
        Else If Year>=MarketFacImplemented then MarketFac = 1;
        Else MarketFac=0;

        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;

        keep State Year TaxRev MarketFac EconNexus;
Run;
```

```
/* Step 2: Merge Combine with control variables (must sort by State Year first) */
proc sort data=Combine; by State Year; run;
proc sort data=PartyControl3; by State Year; run;
proc sort data=GDP3; by State Year; run;
proc sort data=PersonalIncome3; by State Year; run;
proc sort data=Pop3; by State Year; run;
proc sort data=Rate3; by State Year; run;
proc sort data=Threshold3; by State Year; run;

/* Step 3: Sequential merge */
data CombineFull;
        merge Combine
            PartyControl3
            GDP3
            PersonalIncome3
            Pop3
            Rate3
            Threshold3;
        by State Year;
run;

/*Model 3 - TaxRev w/ Control Variables*/
ods output ParameterEstimates=PEforModel3 DataSummary=ObsModel3
                FitStatistics=AdjRsqModel3 Effects=OverallSigModel3;
proc surveyreg data=CombineFull plots=none;
  class State Year / ref=first;
  cluster State;
```

```
  model TaxRev = MarketFac EconNexus State Year PartyControl GDP PersonalIncome Pop
Rate Threshold / solution adjrsq;
run;
quit;

/*Model 4 - LogTaxRev w/ Control Variables*/
data LogCombineFull;
        set CombineFull;
        LogTaxRev = Log(TaxRev);
Run;

ods output ParameterEstimates=PEforModel4 DataSummary=ObsModel4
                FitStatistics=AdjRsqModel4 Effects=OverallSigModel4;
proc surveyreg data=LogCombineFull plots=none;
  class State Year / ref=first;
  cluster State;
  model LogTaxRev = MarketFac EconNexus State Year  PartyControl GDP PersonalIncome
Pop Rate Threshold / solution adjrsq;
run;
quit;

/*PER-CAPITA MODELS*/

/*Combine*/
Data CombinePop;
        Merge TaxRevPop3 Implemented;
        By State;
```

```
	/*MarketFac DID*/
	If MarketFacImplemented="-"  then MarketFac = 0;
	Else If Year>=MarketFacImplemented then MarketFac = 1;
	Else MarketFac=0;

	/*EconNexus DID*/
	If EconNexusImplemented="-" then EconNexus = 0;
	Else If Year>=EconNexusImplemented then EconNexus = 1;
	Else EconNexus=0;

	Keep State Year TaxRevPop MarketFac EconNexus;
Run;

/*Model 5 - TaxRevPop w/o Control Variables*/
ods output ParameterEstimates=PEforModel5 DataSummary=ObsModel5
		FitStatistics=AdjRsqModel5 Effects=OverallSigModel5;
proc surveyreg data=CombinePop plots=none;
  class State Year / ref=first;
  cluster State;
  model TaxRevPop = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

Data test;
	set CombinePop;
run;
```

```
/*Model 6 - LogTaxRevPop w/o Control Variables*/
data LogCombinePop;
	set CombinePop;
	LogTaxRevPop = Log(TaxRevPop);
Run;

ods output ParameterEstimates=PEforModel6 DataSummary=ObsModel6
		FitStatistics=AdjRsqModel6 Effects=OverallSigModel6;
proc surveyreg data=LogCombinePop plots=none;
  class State Year / ref=first;
  cluster State;
  model LogTaxRevPop = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Control Variables*/
Data CombinePop2;
	Merge TaxRevPop3 Implemented;
	By State;

	If MarketFacImplemented="-"  then MarketFac = 0;
	Else If Year>=MarketFacImplemented then MarketFac = 1;
	Else MarketFac=0;

	If EconNexusImplemented="-" then EconNexus = 0;
	Else If Year>=EconNexusImplemented then EconNexus = 1;
	Else EconNexus=0;
```

```
        keep State Year TaxRevPop MarketFac EconNexus PartyControl;
Run;

/* Step 2: Merge Combine with control variables (must sort by State Year first) */
proc sort data=CombinePop2; by State Year; run;
proc sort data=PartyControl3; by State Year; run;
proc sort data=GDPPop3; by State Year; run;
proc sort data=PersonalIncomePop3; by State Year; run;
proc sort data=Rate3; by State Year; run;
proc sort data=Threshold3; by State Year; run;

/* Step 3: Sequential merge */
data CombineFullPop;
        merge CombinePop2
            PartyControl3
            GDPPop3
            PersonalIncomePop3
            Rate3
            Threshold3;
        by State Year;
run;

/*Model 7 - TaxRevPop w/ Control Variables*/
ods output ParameterEstimates=PEforModel7 DataSummary=ObsModel7
                FitStatistics=AdjRsqModel7 Effects=OverallSigModel7;
proc surveyreg data=CombineFullPop plots=none;
```

```
  class State Year / ref=first;
  cluster State;
  model TaxRevPop = MarketFac EconNexus State Year PartyControl GDPPop
PersonalIncomePop Rate Threshold / solution adjrsq;
run;
quit;

/*Model 8 - LogTaxRevPop w/ Control Variables*/
data LogCombineFullPop;
        set CombineFullPop;
        LogTaxRevPop = Log(TaxRevPop);
Run;

ods output ParameterEstimates=PEforModel8 DataSummary=ObsModel8
                FitStatistics=AdjRsqModel8 Effects=OverallSigModel8;
proc surveyreg data=LogCombineFullPop plots=none;
  class State Year / ref=first;
  cluster State;
  model LogTaxRevPop = MarketFac EconNexus State Year PartyControl GDPPop
PersonalIncomePop Rate Threshold / solution adjrsq;
run;
quit;

/*Cleaning*/
Data Table_Long;
        length Model $10;
        length Parameter $30;
```

```
set PEforModel1 PEforModel2 PEforModel3 PEforModel4 PEforModel5 PEforModel6
PEforModel7 PEforModel8 indsname=M;
ThisIsM=M;

length Star $3;
if Probt le 0.01 then Star="***";
    else if Probt le 0.05 then Star="**";
    else if Probt le 0.10 then Star="*";
    else Star="";

if M="WORK.PEFORMODEL1" then Model="Model1";
    else if M="WORK.PEFORMODEL2" then Model="Model2";
    else if M="WORK.PEFORMODEL3" then Model="Model3";
    else if M="WORK.PEFORMODEL4" then Model="Model4";
    else if M="WORK.PEFORMODEL5" then Model="Model5";
    else if M="WORK.PEFORMODEL6" then Model="Model6";
    else if M="WORK.PEFORMODEL7" then Model="Model7";
    else if M="WORK.PEFORMODEL8" then Model="Model8";

Results=Estimate;
EditedResults=Cats(put(Results,comma16.2),Star);
output;

Results=StdErr;
EditedResults=Cats("(",put(Results,comma16.2),")");
output;
```

```
        keep Model Parameter EditedResults /*Probt Star*/;

run;

proc sort data=Table_Long out=Table_Long_Sorted;
        by Model Parameter;
run;

/* Split cleaned estimates into individual model datasets */
data Model1Results(rename=(EditedResults=Model1));
   set Table_Long_Sorted;
   if Model = "Model1";
   drop Model;
run;

data Model2Results(rename=(EditedResults=Model2));
   set Table_Long_Sorted;
   if Model = "Model2";
   drop Model;
run;

data Model3Results(rename=(EditedResults=Model3));
   set Table_Long_Sorted;
   if Model = "Model3";
   drop Model;
run;
```

```
data Model4Results(rename=(EditedResults=Model4));
    set Table_Long_Sorted;
    if Model = "Model4";
    drop Model;
run;

data Model5Results(rename=(EditedResults=Model5));
    set Table_Long_Sorted;
    if Model = "Model5";
    drop Model;
run;

data Model6Results(rename=(EditedResults=Model6));
    set Table_Long_Sorted;
    if Model = "Model6";
    drop Model;
run;

data Model7Results(rename=(EditedResults=Model7));
    set Table_Long_Sorted;
    if Model = "Model7";
    drop Model;
run;

data Model8Results(rename=(EditedResults=Model8));
    set Table_Long_Sorted;
    if Model = "Model8";
```

```
   drop Model;
run;


data Table_Wide;
        merge Model1Results Model2Results Model3Results Model4Results Model5Results
Model6Results Model7Results Model8Results;
        by Parameter;


        length Order 3;
        if Parameter="week" then Order=1;
                else if Parameter="age" then Order=2;
                else Order=3;


        if mod(_n_,2)=1 then Regressors=Parameter;


run;


proc sort data=Table_Wide out=Table_Wide_Sorted(drop=Order Parameter);
        by Order;
run;


Data Table_Long;
        length Model $10;
        length Parameter $30;
        set PEforModel1 PEforModel2 PEforModel3 PEforModel4 PEforModel5 PEforModel6
PEforModel7 PEforModel8 indsname=M;
        ThisIsM=M;
```

```
Where (Estimate ne 0) and (substr(Parameter,1,5) ne "State") and (substr(Parameter,1,4)
ne "Year");

length Star $3;
if Probt le 0.01 then Star="***";
        else if Probt le 0.05 then Star="**";
        else if Probt le 0.10 then Star="*";
        else Star="";

if M="WORK.PEFORMODEL1" then Model="Model1";
        else if M="WORK.PEFORMODEL2" then Model="Model2";
        else if M="WORK.PEFORMODEL3" then Model="Model3";
        else if M="WORK.PEFORMODEL4" then Model="Model4";
        else if M="WORK.PEFORMODEL5" then Model="Model5";
        else if M="WORK.PEFORMODEL6" then Model="Model6";
        else if M="WORK.PEFORMODEL7" then Model="Model7";
        else if M="WORK.PEFORMODEL8" then Model="Model8";

Results=Estimate;
EditedResults=Cats(put(Results,comma16.4),Star);
output;

Results=StdErr;
EditedResults=Cats("(",put(Results,comma16.4),")");
output;

keep Model Parameter EditedResults /*Probt Star*/;
```

```
run;

proc sort data=Table_Long out=Table_Long_Sorted;
        by Model Parameter;
run;

data _null_;
   put "STARTING CLEAN SPLIT OF MODEL RESULTS";
run;

data
   Model1Results(rename=(EditedResults=Model1))
   Model2Results(rename=(EditedResults=Model2))
   Model3Results(rename=(EditedResults=Model3))
   Model4Results(rename=(EditedResults=Model4))
   Model5Results(rename=(EditedResults=Model5))
   Model6Results(rename=(EditedResults=Model6))
   Model7Results(rename=(EditedResults=Model7))
   Model8Results(rename=(EditedResults=Model8));

   set Table_Long_Sorted;

   select (Model);
      when ("Model1") output Model1Results;
      when ("Model2") output Model2Results;
      when ("Model3") output Model3Results;
```

```
        when ("Model4") output Model4Results;
        when ("Model5") output Model5Results;
        when ("Model6") output Model6Results;
        when ("Model7") output Model7Results;
        when ("Model8") output Model8Results;
        otherwise;
    end;

    drop Model;
run;

/*Ordering*/
data Table_Wide;
        merge Model1Results Model2Results Model3Results Model4Results Model5Results
Model6Results Model7Results Model8Results;
        by Parameter;

        length Order 3;
        if Parameter="DID" then order=1;
                else if Parameter="State" then order=2;
                else if Parameter="Year" then order=3;
                else if Parameter="Intercept" then order=4;

        if mod(_n_,2)=1 then Regressors=Parameter;

run;
```

```
proc sort data=Table_Wide out=Table_Wide_Sorted(drop=Order Parameter);
	by Order;
run;

/*Rows for other statistics and info*/
/*Number of Obs.*/

/* Filter and format number of observations for each model */
data Obs1; set ObsModel1; if Label1="Number of Observations"; Model1 = put(NValue1, comma16.); keep Label1 Model1; run;
data Obs2; set ObsModel2; if Label1="Number of Observations"; Model2 = put(NValue1, comma16.); keep Label1 Model2; run;
data Obs3; set ObsModel3; if Label1="Number of Observations"; Model3 = put(NValue1, comma16.); keep Label1 Model3; run;
data Obs4; set ObsModel4; if Label1="Number of Observations"; Model4 = put(NValue1, comma16.); keep Label1 Model4; run;
data Obs5; set ObsModel5; if Label1="Number of Observations"; Model5 = put(NValue1, comma16.); keep Label1 Model5; run;
data Obs6; set ObsModel6; if Label1="Number of Observations"; Model6 = put(NValue1, comma16.); keep Label1 Model6; run;
data Obs7; set ObsModel7; if Label1="Number of Observations"; Model7 = put(NValue1, comma16.); keep Label1 Model7; run;
data Obs8; set ObsModel8; if Label1="Number of Observations"; Model8 = put(NValue1, comma16.); keep Label1 Model8; run;

/* Merge all */
data NumofObs;
	merge Obs1 Obs2 Obs3 Obs4 Obs5 Obs6 Obs7 Obs8;
	by Label1;
run;
```

```
/*Adjusted R-Square*/

/* Filter and rename each Adjusted R-Square dataset individually */
data AdjRsq1; set AdjRsqModel1 (rename=(CValue1=Model1) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq2; set AdjRsqModel2 (rename=(CValue1=Model2) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq3; set AdjRsqModel3 (rename=(CValue1=Model3) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq4; set AdjRsqModel4 (rename=(CValue1=Model4) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq5; set AdjRsqModel5 (rename=(CValue1=Model5) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq6; set AdjRsqModel6 (rename=(CValue1=Model6) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq7; set AdjRsqModel7 (rename=(CValue1=Model7) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq8; set AdjRsqModel8 (rename=(CValue1=Model8) drop=NValue1); if
Label1="Adjusted R-Square"; run;

/* Merge all together */
data AdjRsq;
        merge AdjRsq1 AdjRsq2 AdjRsq3 AdjRsq4 AdjRsq5 AdjRsq6 AdjRsq7 AdjRsq8;
        by Label1;
run;

/*Overall Significance (F-Test)*/

/* Step 1: Filter and format each model's F-test result */
```

```
data OSM1; set OverallSigModel1; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model1 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model1;
run;

data OSM2; set OverallSigModel2; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model2 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model2;
run;

data OSM3; set OverallSigModel3; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model3 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model3;
run;

data OSM4; set OverallSigModel4; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model4 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model4;
run;
```

```
data OSM5; set OverallSigModel5; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model5 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model5;
run;

data OSM6; set OverallSigModel6; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model6 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model6;
run;

data OSM7; set OverallSigModel7; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model7 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model7;
run;

data OSM8; set OverallSigModel8; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model8 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model8;
run;
```

```
/* Step 2: Merge all formatted outputs */
data OverallSig;
   merge OSM1 OSM2 OSM3 OSM4 OSM5 OSM6 OSM7 OSM8;
   by Label1;
run;

/*Combine F-values*/
Data OverallSig;
        merge OSM1 OSM2 OSM3 OSM4 OSM5 OSM6 OSM7 OSM8;
run;

/*Controls*/
Data Controls;
        length Label1 $30;

        Label1="Controls for State?";
        Model1="Yes"; Model2="Yes"; Model3="Yes"; Model4="Yes";
        Model5="Yes"; Model6="Yes"; Model7="Yes"; Model8="Yes";
        output;

        Label1="Controls for Year?";
        output;
run;

/*Combine Rows*/
Data OtherStat;
        set NumofObs AdjRsq OverallSig Controls;
```

```
        rename Label1=Regressors;
run;

/*Add Other Statistics to Results Table*/

Data Table_wide_sorted_withstat;
        set Table_Wide_Sorted OtherStat;
run;

/* Print Cleaned Table*/
ods excel file="/home/u63724901/MySAS/AllModelsCleanedHRP_TaxRev.xlsx"
options(Embedded_Titles="ON" Embedded_Footnotes="ON");
Title "Title";
footnote1 justify=left "Sources: Source";
footnote2 justify=left "Notes: robust standard errors are in parentheses. *, **, and ***
indicate 10%, 5%, and 1% significance levels, respectively.";
proc print data=Table_Wide_Sorted_withstat noobs;
        var regressors;
        var Model1-Model8 / Style(header)={just=center} style(data)={just=center};
        format Regressors $VariableName.;
run;
ods excel close;

/*Summary Stats*/
ods excel file="/home/u63724901/MySAS/TaxRevPop_Summary_All.xlsx"
options(sheet_interval='none');

/* Sheet 1: Overall Summary */
```

```
ods excel options(sheet_name='Overall Summary');
proc means data=TaxRevPop3 n mean median std min max;
   var TaxRevPop;
run;

/* Sheet 2: By State */
ods excel options(sheet_name='By State');
proc means data=TaxRevPop3 n mean median std min max;
   class State;
   var TaxRevPop;
run;

/* Sheet 3: By Year */
ods excel options(sheet_name='By Year');
proc means data=TaxRevPop3 n mean median std min max;
   class Year;
   var TaxRevPop;
run;

ods excel close;
/*/////Entrepreneurship/////*/

/*First Data*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Entrep
        dbms=xlsx
        replace;
```

```
sheet="Entrepreneur_1-4";
getnames=yes;
run;

Proc Sort Data=Entrep;
	by State;
run;

Proc Transpose Data=Entrep Out=Entrep2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data Entrep3;
	Set Entrep2;
	Year=input(_Name_,4.);
	Entrep=Col1;
	keep State Year Entrep;
Run;

/*Second Data*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Implemented
	dbms=xlsx
	replace;
```

```
sheet="Implemented";
getnames=yes;
run;

Proc Sort Data=Implemented;
        by State;
run;

/*Combine*/
Data Combine;
        Merge Entrep3 Implemented;
        By State;

        /*MarketFac DID*/
        If MarketFacImplemented="-"  then MarketFac = 0;
        Else If Year>=MarketFacImplemented then MarketFac = 1;
        Else MarketFac=0;

        /*EconNexus DID*/
        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;

        Keep State Year Entrep MarketFac EconNexus;
Run;

Proc Sort data=Combine;
```

```
        by state year;
run;

/*Model 1 - Entrep w/o Control Variables*/
ods output ParameterEstimates=PEforModel1 DataSummary=ObsModel1
                FitStatistics=AdjRsqModel1 Effects=OverallSigModel1;
proc surveyreg data=Combine plots=none;
  class State Year / ref=first;
  cluster State;
  model Entrep = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Model 2 - LogEntrep w/o Control Variables*/
data LogCombine;
        set Combine;
        LogEntrep = Log(Entrep);
Run;

ods output ParameterEstimates=PEforModel2 DataSummary=ObsModel2
                FitStatistics=AdjRsqModel2 Effects=OverallSigModel2;
proc surveyreg data=LogCombine plots=none;
  class State Year / ref=first;
  cluster State;
  model LogEntrep = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;
```

```
/*Control Variables Uploading*/

/*Third Data - Party Control*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.PartyControl
        dbms=xlsx
        replace;
sheet="PartyControl";
getnames=yes;
run;

Proc Sort Data=PartyControl;
        by State;
run;

Proc Transpose Data=PartyControl Out=PartyControl2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PartyControl3;
        Set PartyControl2;
        Year=input(_Name_,4.);
        PartyControl=Col1;
```

```
        keep State Year PartyControl;
Run;

/*Fourth Data - GDP*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.GDP
        dbms=xlsx
        replace;
sheet="GDP";
getnames=yes;
run;

Proc Sort Data=GDP;
        by State;
run;

Proc Transpose Data=GDP Out=GDP2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data GDP3;
        Set GDP2;
        Year=input(_Name_,4.);
        GDP=Col1;
```

```
        keep State Year GDP;
Run;

/*Fifth Data - Personal Income*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.PersonalIncome
        dbms=xlsx
        replace;
sheet="PersonalIncome";
getnames=yes;
run;

Proc Sort Data=PersonalIncome;
        by State;
run;

Proc Transpose Data=PersonalIncome Out=PersonalIncome2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PersonalIncome3;
        Set PersonalIncome2;
        Year=input(_Name_,4.);
        PersonalIncome=Col1;
```

```
        keep State Year PersonalIncome;
Run;

/*Sixth Data - Population*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Pop
        dbms=xlsx
        replace;
sheet="Pop";
getnames=yes;
run;

Proc Sort Data=Pop;
        by State;
run;

Proc Transpose Data=Pop Out=Pop2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data Pop3;
        Set Pop2;
        Year=input(_Name_,4.);
        Pop=Col1;
```

```
        keep State Year Pop;
Run;

/*Seventh Data - Entrep/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.EntrepPop
        dbms=xlsx
        replace;
sheet="EntrepreneurPop_1-4";
getnames=yes;
run;

Proc Sort Data=EntrepPop;
        by State;
run;

Proc Transpose Data=EntrepPop Out=EntrepPop2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

/*Data EntrepPop3;
        Set EntrepPop2;
        Year=input(_Name_,4.);
        EntrepPop=Col1;
```

```
        keep State Year EntrepPop;
Run;*/

Data EntrepPop3;
        Set EntrepPop2;
        Year = input(_Name_, 4.);
        EntrepPop = Col1 * 10000; /* redefine in units per 10,000 residents */
        keep State Year EntrepPop;
Run;

/*Eighth Data - GDP/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.GDPPop
        dbms=xlsx
        replace;
sheet="GDPPop";
getnames=yes;
run;

Proc Sort Data=GDPPop;
        by State;
run;

Proc Transpose Data=GDPPop Out=GDPPop2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
```

```
"2020"n "2021"n "2022"n;
Run;

Data GDPPop3;
        Set GDPPop2;
        Year=input(_Name_,4.);
        GDPPop=Col1;
        keep State Year GDPPop;
Run;

/*Ninth Data - Personal Income/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.PersonalIncomePop
        dbms=xlsx
        replace;
sheet="PersIncPop";
getnames=yes;
run;

Proc Sort Data=PersonalIncomePop;
        by State;
run;

Proc Transpose Data=PersonalIncomePop Out=PersonalIncomePop2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
```

```
"2020"n "2021"n "2022"n;
Run;

Data PersonalIncomePop3;
        Set PersonalIncomePop2;
        Year=input(_Name_,4.);
        PersonalIncomePop=Col1;
        keep State Year PersonalIncomePop;
Run;

/*Tenth Data - Tax Rates*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Rate
        dbms=xlsx
        replace;
sheet="Rate";
getnames=yes;
run;

Proc Sort Data=Rate;
        by State;
run;

Proc Transpose Data=Rate Out=Rate2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
```

```
"2020"n "2021"n "2022"n "2023"n;
Run;

Data Rate3;
        Set Rate2;
        Year=input(_Name_,4.);
        Rate=Col1;
        keep State Year Rate;
Run;

/*Eleventh Data - EN Threshold*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Threshold
        dbms=xlsx
        replace;
sheet="Threshold";
getnames=yes;
run;

Proc Sort Data=Threshold;
        by State;
run;

Proc Transpose Data=Threshold Out=Threshold2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
```

```
"2020"n "2021"n "2022"n "2023"n;
Run;

Data Threshold3;
	Set Threshold2;
	Year=input(_Name_,4.);
	Threshold=Col1;
	keep State Year Threshold;
Run;

/*NEW MODELS*/

/*Combining Everything*/

/* Step 1: Merge Entrep3 with Implemented (already done) */
Data Combine;
	Merge Entrep3 Implemented;
	By State;

	If MarketFacImplemented="-"  then MarketFac = 0;
	Else If Year>=MarketFacImplemented then MarketFac = 1;
	Else MarketFac=0;

	If EconNexusImplemented="-" then EconNexus = 0;
	Else If Year>=EconNexusImplemented then EconNexus = 1;
	Else EconNexus=0;
```

```
        keep State Year Entrep MarketFac EconNexus;
Run;

/* Step 2: Merge Combine with control variables (must sort by State Year first) */
proc sort data=Combine; by State Year; run;
proc sort data=PartyControl3; by State Year; run;
proc sort data=GDP3; by State Year; run;
proc sort data=PersonalIncome3; by State Year; run;
proc sort data=Pop3; by State Year; run;
proc sort data=Rate3; by State Year; run;
proc sort data=Threshold3; by State Year; run;

/* Step 3: Sequential merge */
data CombineFull;
        merge Combine
            PartyControl3
            GDP3
            PersonalIncome3
            Pop3
            Rate3
            Threshold3;
        by State Year;
run;

/*Model 3 - Entrep w/ Control Variables*/
ods output ParameterEstimates=PEforModel3 DataSummary=ObsModel3
                FitStatistics=AdjRsqModel3 Effects=OverallSigModel3;
```

```
proc surveyreg data=CombineFull plots=none;
  class State Year / ref=first;
  cluster State;
  model Entrep = MarketFac EconNexus State Year PartyControl GDP PersonalIncome Pop Rate
Threshold / solution adjrsq;
run;
quit;

/*Model 4 - LogEntrep w/ Control Variables*/
data LogCombineFull;
        set CombineFull;
        LogEntrep = Log(Entrep);
Run;

ods output ParameterEstimates=PEforModel4 DataSummary=ObsModel4
                FitStatistics=AdjRsqModel4 Effects=OverallSigModel4;
proc surveyreg data=LogCombineFull plots=none;
  class State Year / ref=first;
  cluster State;
  model LogEntrep = MarketFac EconNexus State Year  PartyControl GDP PersonalIncome Pop
Rate Threshold / solution adjrsq;
run;
quit;

/*PER-CAPITA MODELS*/

/*Combine*/
Data CombinePop;
```

```
	Merge EntrepPop3 Implemented;
	By State;

	/*MarketFac DID*/
	If MarketFacImplemented="-"  then MarketFac = 0;
	Else If Year>=MarketFacImplemented then MarketFac = 1;
	Else MarketFac=0;

	/*EconNexus DID*/
	If EconNexusImplemented="-" then EconNexus = 0;
	Else If Year>=EconNexusImplemented then EconNexus = 1;
	Else EconNexus=0;

	Keep State Year EntrepPop MarketFac EconNexus;
Run;

/*Model 5 - EntrepPop w/o Control Variables*/
ods output ParameterEstimates=PEforModel5 DataSummary=ObsModel5
		FitStatistics=AdjRsqModel5 Effects=OverallSigModel5;
proc surveyreg data=CombinePop plots=none;
  class State Year / ref=first;
  cluster State;
  model EntrepPop = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Model 6 - LogEntrepPop w/o Control Variables*/
```

```
data LogCombinePop;
        set CombinePop;
        LogEntrepPop = Log(EntrepPop);
Run;

ods output ParameterEstimates=PEforModel6 DataSummary=ObsModel6
                FitStatistics=AdjRsqModel6 Effects=OverallSigModel6;
proc surveyreg data=LogCombinePop plots=none;
  class State Year / ref=first;
  cluster State;
  model LogEntrepPop = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Control Variables*/
Data CombinePop2;
        Merge EntrepPop3 Implemented;
        By State;

        If MarketFacImplemented="-"  then MarketFac = 0;
        Else If Year>=MarketFacImplemented then MarketFac = 1;
        Else MarketFac=0;

        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;
```

```
        keep State Year EntrepPop MarketFac EconNexus PartyControl;
Run;

/* Step 2: Merge Combine with control variables (must sort by State Year first) */
proc sort data=CombinePop2; by State Year; run;
proc sort data=PartyControl3; by State Year; run;
proc sort data=GDPPop3; by State Year; run;
proc sort data=PersonalIncomePop3; by State Year; run;
proc sort data=Rate3; by State Year; run;
proc sort data=Threshold3; by State Year; run;

/* Step 3: Sequential merge */
data CombineFullPop;
        merge CombinePop2
            PartyControl3
            GDPPop3
            PersonalIncomePop3
            Rate3
            Threshold3;
        by State Year;
run;

/*Model 7 - EntrepPop w/ Control Variables*/
ods output ParameterEstimates=PEforModel7 DataSummary=ObsModel7
                FitStatistics=AdjRsqModel7 Effects=OverallSigModel7;
proc surveyreg data=CombineFullPop plots=none;
  class State Year / ref=first;
```

```
  cluster State;
  model EntrepPop = MarketFac EconNexus State Year PartyControl GDPPop
PersonalIncomePop Rate Threshold / solution adjrsq;
run;
quit;

/*Model 8 - LogEntrepPop w/ Control Variables*/
data LogCombineFullPop;
        set CombineFullPop;
        LogEntrepPop = Log(EntrepPop);
Run;

ods output ParameterEstimates=PEforModel8 DataSummary=ObsModel8
                FitStatistics=AdjRsqModel8 Effects=OverallSigModel8;
proc surveyreg data=LogCombineFullPop plots=none;
  class State Year / ref=first;
  cluster State;
  model LogEntrepPop = MarketFac EconNexus State Year PartyControl GDPPop
PersonalIncomePop Rate Threshold / solution adjrsq;
run;
quit;

/*Cleaning*/
Data Table_Long;
        length Model $10;
        length Parameter $30;
        set PEforModel1 PEforModel2 PEforModel3 PEforModel4 PEforModel5 PEforModel6
PEforModel7 PEforModel8 indsname=M;
```

```
ThisIsM=M;

length Star $3;
if Probt le 0.01 then Star="***";
        else if Probt le 0.05 then Star="**";
        else if Probt le 0.10 then Star="*";
        else Star="";

if M="WORK.PEFORMODEL1" then Model="Model1";
        else if M="WORK.PEFORMODEL2" then Model="Model2";
        else if M="WORK.PEFORMODEL3" then Model="Model3";
        else if M="WORK.PEFORMODEL4" then Model="Model4";
        else if M="WORK.PEFORMODEL5" then Model="Model5";
        else if M="WORK.PEFORMODEL6" then Model="Model6";
        else if M="WORK.PEFORMODEL7" then Model="Model7";
        else if M="WORK.PEFORMODEL8" then Model="Model8";

Results=Estimate;
EditedResults=Cats(put(Results,comma16.2),Star);
output;

Results=StdErr;
EditedResults=Cats("(",put(Results,comma16.2),")");
output;

keep Model Parameter EditedResults /*Probt Star*/;
```

```
run;

proc sort data=Table_Long out=Table_Long_Sorted;
        by Model Parameter;
run;

/* Split cleaned estimates into individual model datasets */
data Model1Results(rename=(EditedResults=Model1));
   set Table_Long_Sorted;
   if Model = "Model1";
   drop Model;
run;

data Model2Results(rename=(EditedResults=Model2));
   set Table_Long_Sorted;
   if Model = "Model2";
   drop Model;
run;

data Model3Results(rename=(EditedResults=Model3));
   set Table_Long_Sorted;
   if Model = "Model3";
   drop Model;
run;

data Model4Results(rename=(EditedResults=Model4));
   set Table_Long_Sorted;
```

```
    if Model = "Model4";
    drop Model;
run;

data Model5Results(rename=(EditedResults=Model5));
    set Table_Long_Sorted;
    if Model = "Model5";
    drop Model;
run;

data Model6Results(rename=(EditedResults=Model6));
    set Table_Long_Sorted;
    if Model = "Model6";
    drop Model;
run;

data Model7Results(rename=(EditedResults=Model7));
    set Table_Long_Sorted;
    if Model = "Model7";
    drop Model;
run;

data Model8Results(rename=(EditedResults=Model8));
    set Table_Long_Sorted;
    if Model = "Model8";
    drop Model;
run;
```

```
data Table_Wide;
	merge Model1Results Model2Results Model3Results Model4Results Model5Results
Model6Results Model7Results Model8Results;
	by Parameter;


	length Order 3;
	if Parameter="week" then Order=1;
		else if Parameter="age" then Order=2;
		else Order=3;


	if mod(_n_,2)=1 then Regressors=Parameter;


run;


proc sort data=Table_Wide out=Table_Wide_Sorted(drop=Order Parameter);
	by Order;
run;


Data Table_Long;
	length Model $10;
	length Parameter $30;
	set PEforModel1 PEforModel2 PEforModel3 PEforModel4 PEforModel5 PEforModel6
PEforModel7 PEforModel8 indsname=M;
	ThisIsM=M;
	Where (Estimate ne 0) and (substr(Parameter,1,5) ne "State") and (substr(Parameter,1,4)
ne "Year");
```

```
        length Star $3;
        if Probt le 0.01 then Star="***";
                else if Probt le 0.05 then Star="**";
                else if Probt le 0.10 then Star="*";
                else Star="";

        if M="WORK.PEFORMODEL1" then Model="Model1";
                else if M="WORK.PEFORMODEL2" then Model="Model2";
                else if M="WORK.PEFORMODEL3" then Model="Model3";
                else if M="WORK.PEFORMODEL4" then Model="Model4";
                else if M="WORK.PEFORMODEL5" then Model="Model5";
                else if M="WORK.PEFORMODEL6" then Model="Model6";
                else if M="WORK.PEFORMODEL7" then Model="Model7";
                else if M="WORK.PEFORMODEL8" then Model="Model8";

        Results=Estimate;
        EditedResults=Cats(put(Results,comma16.4),Star);
        output;

        Results=StdErr;
        EditedResults=Cats("(",put(Results,comma16.4),")");
        output;

        keep Model Parameter EditedResults /*Probt Star*/;

run;
```

```
proc sort data=Table_Long out=Table_Long_Sorted;
        by Model Parameter;
run;

data _null_;
   put "STARTING CLEAN SPLIT OF MODEL RESULTS";
run;

data
   Model1Results(rename=(EditedResults=Model1))
   Model2Results(rename=(EditedResults=Model2))
   Model3Results(rename=(EditedResults=Model3))
   Model4Results(rename=(EditedResults=Model4))
   Model5Results(rename=(EditedResults=Model5))
   Model6Results(rename=(EditedResults=Model6))
   Model7Results(rename=(EditedResults=Model7))
   Model8Results(rename=(EditedResults=Model8));

   set Table_Long_Sorted;

   select (Model);
      when ("Model1") output Model1Results;
      when ("Model2") output Model2Results;
      when ("Model3") output Model3Results;
      when ("Model4") output Model4Results;
      when ("Model5") output Model5Results;
      when ("Model6") output Model6Results;
```

```
        when ("Model7") output Model7Results;
        when ("Model8") output Model8Results;
        otherwise;
    end;

    drop Model;
run;

/*Ordering*/
data Table_Wide;
        merge Model1Results Model2Results Model3Results Model4Results Model5Results
Model6Results Model7Results Model8Results;
        by Parameter;

        length Order 3;
        if Parameter="DID" then order=1;
                else if Parameter="State" then order=2;
                else if Parameter="Year" then order=3;
                else if Parameter="Intercept" then order=4;

        if mod(_n_,2)=1 then Regressors=Parameter;

run;

proc sort data=Table_Wide out=Table_Wide_Sorted(drop=Order Parameter);
        by Order;
run;
```

```
/*Rows for other statistics and info*/

/*Number of Obs.*/


/* Filter and format number of observations for each model */

data Obs1; set ObsModel1; if Label1="Number of Observations"; Model1 = put(NValue1,
comma16.); keep Label1 Model1; run;

data Obs2; set ObsModel2; if Label1="Number of Observations"; Model2 = put(NValue1,
comma16.); keep Label1 Model2; run;

data Obs3; set ObsModel3; if Label1="Number of Observations"; Model3 = put(NValue1,
comma16.); keep Label1 Model3; run;

data Obs4; set ObsModel4; if Label1="Number of Observations"; Model4 = put(NValue1,
comma16.); keep Label1 Model4; run;

data Obs5; set ObsModel5; if Label1="Number of Observations"; Model5 = put(NValue1,
comma16.); keep Label1 Model5; run;

data Obs6; set ObsModel6; if Label1="Number of Observations"; Model6 = put(NValue1,
comma16.); keep Label1 Model6; run;

data Obs7; set ObsModel7; if Label1="Number of Observations"; Model7 = put(NValue1,
comma16.); keep Label1 Model7; run;

data Obs8; set ObsModel8; if Label1="Number of Observations"; Model8 = put(NValue1,
comma16.); keep Label1 Model8; run;


/* Merge all */

data NumofObs;

        merge Obs1 Obs2 Obs3 Obs4 Obs5 Obs6 Obs7 Obs8;

        by Label1;

run;


/*Adjusted R-Square*/
```

```
/* Filter and rename each Adjusted R-Square dataset individually */

data AdjRsq1; set AdjRsqModel1 (rename=(CValue1=Model1) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq2; set AdjRsqModel2 (rename=(CValue1=Model2) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq3; set AdjRsqModel3 (rename=(CValue1=Model3) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq4; set AdjRsqModel4 (rename=(CValue1=Model4) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq5; set AdjRsqModel5 (rename=(CValue1=Model5) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq6; set AdjRsqModel6 (rename=(CValue1=Model6) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq7; set AdjRsqModel7 (rename=(CValue1=Model7) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq8; set AdjRsqModel8 (rename=(CValue1=Model8) drop=NValue1); if
Label1="Adjusted R-Square"; run;


/* Merge all together */

data AdjRsq;

        merge AdjRsq1 AdjRsq2 AdjRsq3 AdjRsq4 AdjRsq5 AdjRsq6 AdjRsq7 AdjRsq8;

        by Label1;

run;


/*Overall Significance (F-Test)*/


/* Step 1: Filter and format each model's F-test result */

data OSM1; set OverallSigModel1; if Effect="Model"; Label1="Overall Significance";

   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
```

```
   Model1 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model1;
run;

data OSM2; set OverallSigModel2; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
   Model2 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model2;
run;

data OSM3; set OverallSigModel3; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
   Model3 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model3;
run;

data OSM4; set OverallSigModel4; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
   Model4 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model4;
run;

data OSM5; set OverallSigModel5; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
```

```
    Model5 = cats(put(Fvalue, comma20.2), Star);
    keep Label1 Model5;
run;

data OSM6; set OverallSigModel6; if Effect="Model"; Label1="Overall Significance";
    length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
    Model6 = cats(put(Fvalue, comma20.2), Star);
    keep Label1 Model6;
run;

data OSM7; set OverallSigModel7; if Effect="Model"; Label1="Overall Significance";
    length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
    Model7 = cats(put(Fvalue, comma20.2), Star);
    keep Label1 Model7;
run;

data OSM8; set OverallSigModel8; if Effect="Model"; Label1="Overall Significance";
    length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
    Model8 = cats(put(Fvalue, comma20.2), Star);
    keep Label1 Model8;
run;

/* Step 2: Merge all formatted outputs */
data OverallSig;
    merge OSM1 OSM2 OSM3 OSM4 OSM5 OSM6 OSM7 OSM8;
```

```
   by Label1;
run;

/*Combine F-values*/
Data OverallSig;
        merge OSM1 OSM2 OSM3 OSM4 OSM5 OSM6 OSM7 OSM8;
run;

/*Controls*/
Data Controls;
        length Label1 $30;

        Label1="Controls for State?";
        Model1="Yes"; Model2="Yes"; Model3="Yes"; Model4="Yes";
        Model5="Yes"; Model6="Yes"; Model7="Yes"; Model8="Yes";
        output;

        Label1="Controls for Year?";
        output;
run;

/*Combine Rows*/
Data OtherStat;
        set NumofObs AdjRsq OverallSig Controls;
        rename Label1=Regressors;
run;
```

```
/*Add Other Statistics to Results Table*/

Data Table_wide_sorted_withstat;
	set Table_Wide_Sorted OtherStat;
run;

/* Print Cleaned Table*/
ods excel file="/home/u63724901/MySAS/AllModelsCleanedHRP_Entrep_1-4.xlsx"
options(Embedded_Titles="ON" Embedded_Footnotes="ON");
Title "Title";
footnote1 justify=left "Sources: Source";
footnote2 justify=left "Notes: robust standard errors are in parentheses. *, **, and ***
indicate 10%, 5%, and 1% significance levels, respectively.";
proc print data=Table_Wide_Sorted_withstat noobs;
	var regressors;
	var Model1-Model8 / Style(header)={just=center} style(data)={just=center};
	format Regressors $VariableName.;
run;
ods excel close;

/*//////////////Parallel Trends/////////////////*/
/* STEP 1: Merge EntrepPop with Implemented */
data CombineEntrepPop;
   merge EntrepPop3 Implemented;
   by State;

   /* MarketFac & EconNexus DIDs */
```

```
    if MarketFacImplemented="-" then MarketFac = 0;
    else if Year >= MarketFacImplemented then MarketFac = 1;
    else MarketFac = 0;

    if EconNexusImplemented="-" then EconNexus = 0;
    else if Year >= EconNexusImplemented then EconNexus = 1;
    else EconNexus = 0;
run;

/* STEP 2: Assign treatment and relative time */
data CombineEntrepPop2;
    set CombineEntrepPop;

    /* Control group: states without sales tax */
    if State in (2, 8, 26, 29, 37) then treatment = 0;
    else treatment = 1;

    /* Relative time from 2018 */
    rt = Year - 2018;
run;

Data CombineEntrepPop3;
        set CombineEntrepPop2;
        EntrepPop10k=EntrepPop*10000;
run;

/*proc surveyreg data=Test;
```

```
   where Year < 2018;
   /*class State;*/
   /*model Test = rt rt*rt rt*rt*rt
            treatment treatment*rt treatment*rt*rt treatment*rt*rt*rt
            /*State*/
            /*/ solution adjrsq;
run;*/

/* STEP 3: Run parallel trends test (pre-treatment only) */
ods output ParameterEstimates=PE_ParallelTrends_EntrepPop
      FitStatistics=Fit_ParallelTrends_EntrepPop;

proc surveyreg data=CombineEntrepPop3;
   where Year < 2018;
   /*class State;*/
   model EntrepPop10k = rt rt*rt rt*rt*rt
            treatment treatment*rt treatment*rt*rt treatment*rt*rt*rt
            /*State*/
            / solution adjrsq;
run;

ods output close;

/* STEP 4: Export to Excel */
ods excel file="/home/u63724901/MySAS/ParallelTrends_EntrepPop_1-4.xlsx";

ods excel options(sheet_name="Parameter Estimates");
```

```
proc print data=PE_ParallelTrends_EntrepPop noobs label;
   title "Parallel Trends Test – Per Capita Entrep";
run;

ods excel options(sheet_name="Fit Statistics");
proc print data=Fit_ParallelTrends_EntrepPop noobs label;
   title "Fit Statistics – Per Capita Entrep";
run;

ods excel close;

/*//////////////////Summary Stats////////////*/
ods excel file="/home/u63724901/MySAS/EntrepPop_Summary_All_1-4.xlsx"
options(sheet_interval='none');

/* Sheet 1: Overall Summary */
ods excel options(sheet_name='Overall Summary');
proc means data=EntrepPop3 n mean median std min max;
   var EntrepPop;
run;

/* Sheet 2: By State */
ods excel options(sheet_name='By State');
proc means data=EntrepPop3 n mean median std min max;
   class State;
   var EntrepPop;
run;
```

```
/* Sheet 3: By Year */
ods excel options(sheet_name='By Year');
proc means data=EntrepPop3 n mean median std min max;
   class Year;
   var EntrepPop;
run;

ods excel close;

/*Testing*/

proc freq data=EntrepPop;
   tables State;
run;
/*/////Edu/////*/

/*First Data*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Edu
        dbms=xlsx
        replace;
sheet="Edu";
getnames=yes;
run;

Proc Sort Data=Edu;
```

```
        by State;
run;

Proc Transpose Data=Edu Out=Edu2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data Edu3;
        Set Edu2;
        Year=input(_Name_,4.);
        Edu=Col1;
        keep State Year Edu;
Run;

/*Second Data*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Implemented
        dbms=xlsx
        replace;
sheet="Implemented";
getnames=yes;
run;

Proc Sort Data=Implemented;
```

```
        by State;
run;

/*Combine*/
Data Combine;
        Merge Edu3 Implemented;
        By State;

        /*MarketFac DID*/
        If MarketFacImplemented="-"  then MarketFac = 0;
        Else If Year>=MarketFacImplemented then MarketFac = 1;
        Else MarketFac=0;

        /*EconNexus DID*/
        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;

        Keep State Year Edu MarketFac EconNexus;
Run;

/*Model 1 - Edu w/o Control Variables*/
ods output ParameterEstimates=PEforModel1 DataSummary=ObsModel1
                FitStatistics=AdjRsqModel1 Effects=OverallSigModel1;
proc surveyreg data=Combine plots=none;
  class State Year / ref=first;
  cluster State;
```

```
  model Edu = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Model 2 - LogEdu w/o Control Variables*/
data LogCombine;
        set Combine;
        LogEdu = Log(Edu);
Run;

ods output ParameterEstimates=PEforModel2 DataSummary=ObsModel2
                FitStatistics=AdjRsqModel2 Effects=OverallSigModel2;
proc surveyreg data=LogCombine plots=none;
  class State Year / ref=first;
  cluster State;
  model LogEdu = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Control Variables Uploading*/

/*Third Data - Party Control*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.PartyControl
        dbms=xlsx
        replace;
sheet="PartyControl";
```

```
getnames=yes;
run;

Proc Sort Data=PartyControl;
	by State;
run;

Proc Transpose Data=PartyControl Out=PartyControl2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PartyControl3;
	Set PartyControl2;
	Year=input(_Name_,4.);
	PartyControl=Col1;
	keep State Year PartyControl;
Run;

/*Fourth Data - GDP*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.GDP
	dbms=xlsx
	replace;
sheet="GDP";
```

```
getnames=yes;
run;

Proc Sort Data=GDP;
        by State;
run;

Proc Transpose Data=GDP Out=GDP2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data GDP3;
        Set GDP2;
        Year=input(_Name_,4.);
        GDP=Col1;
        keep State Year GDP;
Run;

/*Fifth Data - Personal Income*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.PersonalIncome
        dbms=xlsx
        replace;
sheet="PersonalIncome";
```

```
getnames=yes;
run;

Proc Sort Data=PersonalIncome;
	by State;
run;

Proc Transpose Data=PersonalIncome Out=PersonalIncome2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PersonalIncome3;
	Set PersonalIncome2;
	Year=input(_Name_,4.);
	PersonalIncome=Col1;
	keep State Year PersonalIncome;
Run;

/*Sixth Data - Population*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Pop
	dbms=xlsx
	replace;
sheet="Pop";
```

```
getnames=yes;
run;

Proc Sort Data=Pop;
	by State;
run;

Proc Transpose Data=Pop Out=Pop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data Pop3;
	Set Pop2;
	Year=input(_Name_,4.);
	Pop=Col1;
	keep State Year Pop;
Run;

/*Seventh Data - Edu/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.EduPop
	dbms=xlsx
	replace;
sheet="EduPop";
```

```
getnames=yes;
run;

Proc Sort Data=EduPop;
	by State;
run;

Proc Transpose Data=EduPop Out=EduPop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data EduPop3;
	Set EduPop2;
	Year=input(_Name_,4.);
	EduPop=Col1;
	keep State Year EduPop;
Run;

/*Eighth Data - GDP/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.GDPPop
	dbms=xlsx
	replace;
sheet="GDPPop";
```

```
getnames=yes;
run;

Proc Sort Data=GDPPop;
	by State;
run;

Proc Transpose Data=GDPPop Out=GDPPop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data GDPPop3;
	Set GDPPop2;
	Year=input(_Name_,4.);
	GDPPop=Col1;
	keep State Year GDPPop;
Run;

/*Ninth Data - Personal Income/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.PersonalIncomePop
	dbms=xlsx
	replace;
sheet="PersIncPop";
```

```
getnames=yes;
run;

Proc Sort Data=PersonalIncomePop;
	by State;
run;

Proc Transpose Data=PersonalIncomePop Out=PersonalIncomePop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PersonalIncomePop3;
	Set PersonalIncomePop2;
	Year=input(_Name_,4.);
	PersonalIncomePop=Col1;
	keep State Year PersonalIncomePop;
Run;

/*Tenth Data - Tax Rates*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Rate
	dbms=xlsx
	replace;
sheet="Rate";
```

```
getnames=yes;
run;

Proc Sort Data=Rate;
	by State;
run;

Proc Transpose Data=Rate Out=Rate2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;

Data Rate3;
	Set Rate2;
	Year=input(_Name_,4.);
	Rate=Col1;
	keep State Year Rate;
Run;

/*Eleventh Data - EN Threshold*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Threshold
	dbms=xlsx
	replace;
sheet="Threshold";
```

```
getnames=yes;
run;

Proc Sort Data=Threshold;
        by State;
run;

Proc Transpose Data=Threshold Out=Threshold2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;

Data Threshold3;
        Set Threshold2;
        Year=input(_Name_,4.);
        Threshold=Col1;
        keep State Year Threshold;
Run;

/*NEW MODELS*/

/*Combining Everything*/

/* Step 1: Merge Edu3 with Implemented (already done) */
Data Combine;
```

```
        Merge Edu3 Implemented;
        By State;

        If MarketFacImplemented="-"  then MarketFac = 0;
        Else If Year>=MarketFacImplemented then MarketFac = 1;
        Else MarketFac=0;

        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;

        keep State Year Edu MarketFac EconNexus;
Run;

/* Step 2: Merge Combine with control variables (must sort by State Year first) */
proc sort data=Combine; by State Year; run;
proc sort data=PartyControl3; by State Year; run;
proc sort data=GDP3; by State Year; run;
proc sort data=PersonalIncome3; by State Year; run;
proc sort data=Pop3; by State Year; run;
proc sort data=Rate3; by State Year; run;
proc sort data=Threshold3; by State Year; run;

/* Step 3: Sequential merge */
data CombineFull;
        merge Combine
            PartyControl3
```

```
            GDP3
            PersonalIncome3
            Pop3
            Rate3
            Threshold3;
        by State Year;
run;

/*Model 3 - Edu w/ Control Variables*/
ods output ParameterEstimates=PEforModel3 DataSummary=ObsModel3
                FitStatistics=AdjRsqModel3 Effects=OverallSigModel3;
proc surveyreg data=CombineFull plots=none;
  class State Year / ref=first;
  cluster State;
  model Edu = MarketFac EconNexus State Year PartyControl GDP PersonalIncome Pop Rate
Threshold / solution adjrsq;
run;
quit;

/*Model 4 - LogEdu w/ Control Variables*/
data LogCombineFull;
        set CombineFull;
        LogEdu = Log(Edu);
Run;

ods output ParameterEstimates=PEforModel4 DataSummary=ObsModel4
                FitStatistics=AdjRsqModel4 Effects=OverallSigModel4;
```

```
proc surveyreg data=LogCombineFull plots=none;
  class State Year / ref=first;
  cluster State;
  model LogEdu = MarketFac EconNexus State Year  PartyControl GDP PersonalIncome Pop
Rate Threshold / solution adjrsq;
run;
quit;

/*PER-CAPITA MODELS*/

/*Combine*/
Data CombinePop;
        Merge EduPop3 Implemented;
        By State;

        /*MarketFac DID*/
        If MarketFacImplemented="-"  then MarketFac = 0;
        Else If Year>=MarketFacImplemented then MarketFac = 1;
        Else MarketFac=0;

        /*EconNexus DID*/
        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;

        Keep State Year EduPop MarketFac EconNexus;
Run;
```

```
/*Model 5 - EduPop w/o Control Variables*/
ods output ParameterEstimates=PEforModel5 DataSummary=ObsModel5
                FitStatistics=AdjRsqModel5 Effects=OverallSigModel5;
proc surveyreg data=CombinePop plots=none;
  class State Year / ref=first;
  cluster State;
  model EduPop = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Model 6 - LogEduPop w/o Control Variables*/
data LogCombinePop;
        set CombinePop;
        LogEduPop = Log(EduPop);
Run;

ods output ParameterEstimates=PEforModel6 DataSummary=ObsModel6
                FitStatistics=AdjRsqModel6 Effects=OverallSigModel6;
proc surveyreg data=LogCombinePop plots=none;
  class State Year / ref=first;
  cluster State;
  model LogEduPop = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Control Variables*/
```

```
Data CombinePop2;
        Merge EduPop3 Implemented;
        By State;

        If MarketFacImplemented="-"  then MarketFac = 0;
        Else If Year>=MarketFacImplemented then MarketFac = 1;
        Else MarketFac=0;

        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;

        keep State Year EduPop MarketFac EconNexus PartyControl;
Run;

/* Step 2: Merge Combine with control variables (must sort by State Year first) */
proc sort data=CombinePop2; by State Year; run;
proc sort data=PartyControl3; by State Year; run;
proc sort data=GDPPop3; by State Year; run;
proc sort data=PersonalIncomePop3; by State Year; run;
proc sort data=Rate3; by State Year; run;
proc sort data=Threshold3; by State Year; run;

/* Step 3: Sequential merge */
data CombineFullPop;
        merge CombinePop2
            PartyControl3
```

```
            GDPPop3
            PersonalIncomePop3
            Rate3
            Threshold3;
        by State Year;
run;

/*Model 7 - EduPop w/ Control Variables*/
ods output ParameterEstimates=PEforModel7 DataSummary=ObsModel7
                FitStatistics=AdjRsqModel7 Effects=OverallSigModel7;
proc surveyreg data=CombineFullPop plots=none;
  class State Year / ref=first;
  cluster State;
  model EduPop = MarketFac EconNexus State Year PartyControl GDPPop PersonalIncomePop
Rate Threshold / solution adjrsq;
run;
quit;

/*Model 8 - LogEduPop w/ Control Variables*/
data LogCombineFullPop;
        set CombineFullPop;
        LogEduPop = Log(EduPop);
Run;

ods output ParameterEstimates=PEforModel8 DataSummary=ObsModel8
                FitStatistics=AdjRsqModel8 Effects=OverallSigModel8;
proc surveyreg data=LogCombineFullPop plots=none;
```

```
  class State Year / ref=first;
  cluster State;
  model LogEduPop = MarketFac EconNexus State Year PartyControl GDPPop
PersonalIncomePop Rate Threshold / solution adjrsq;
run;
quit;

/*Cleaning*/
Data Table_Long;
        length Model $10;
        length Parameter $30;
        set PEforModel1 PEforModel2 PEforModel3 PEforModel4 PEforModel5 PEforModel6
PEforModel7 PEforModel8 indsname=M;
        ThisIsM=M;

        length Star $3;
        if Probt le 0.01 then Star="***";
                else if Probt le 0.05 then Star="**";
                else if Probt le 0.10 then Star="*";
                else Star="";

        if M="WORK.PEFORMODEL1" then Model="Model1";
                else if M="WORK.PEFORMODEL2" then Model="Model2";
                else if M="WORK.PEFORMODEL3" then Model="Model3";
                else if M="WORK.PEFORMODEL4" then Model="Model4";
                else if M="WORK.PEFORMODEL5" then Model="Model5";
                else if M="WORK.PEFORMODEL6" then Model="Model6";
                else if M="WORK.PEFORMODEL7" then Model="Model7";
```

```
                else if M="WORK.PEFORMODEL8" then Model="Model8";

        Results=Estimate;
        EditedResults=Cats(put(Results,comma16.2),Star);
        output;

        Results=StdErr;
        EditedResults=Cats("(",put(Results,comma16.2),")");
        output;

        keep Model Parameter EditedResults /*Probt Star*/;

run;

proc sort data=Table_Long out=Table_Long_Sorted;
        by Model Parameter;
run;

/* Split cleaned estimates into individual model datasets */
data Model1Results(rename=(EditedResults=Model1));
   set Table_Long_Sorted;
   if Model = "Model1";
   drop Model;
run;

data Model2Results(rename=(EditedResults=Model2));
   set Table_Long_Sorted;
```

```
    if Model = "Model2";
    drop Model;
run;

data Model3Results(rename=(EditedResults=Model3));
    set Table_Long_Sorted;
    if Model = "Model3";
    drop Model;
run;

data Model4Results(rename=(EditedResults=Model4));
    set Table_Long_Sorted;
    if Model = "Model4";
    drop Model;
run;

data Model5Results(rename=(EditedResults=Model5));
    set Table_Long_Sorted;
    if Model = "Model5";
    drop Model;
run;

data Model6Results(rename=(EditedResults=Model6));
    set Table_Long_Sorted;
    if Model = "Model6";
    drop Model;
run;
```

```
data Model7Results(rename=(EditedResults=Model7));
   set Table_Long_Sorted;
   if Model = "Model7";
   drop Model;
run;

data Model8Results(rename=(EditedResults=Model8));
   set Table_Long_Sorted;
   if Model = "Model8";
   drop Model;
run;

data Table_Wide;
        merge Model1Results Model2Results Model3Results Model4Results Model5Results
Model6Results Model7Results Model8Results;
        by Parameter;

        length Order 3;
        if Parameter="week" then Order=1;
                else if Parameter="age" then Order=2;
                else Order=3;

        if mod(_n_,2)=1 then Regressors=Parameter;

run;
```

```
proc sort data=Table_Wide out=Table_Wide_Sorted(drop=Order Parameter);
	by Order;
run;


Data Table_Long;
	length Model $10;
	length Parameter $30;
	set PEforModel1 PEforModel2 PEforModel3 PEforModel4 PEforModel5 PEforModel6
PEforModel7 PEforModel8 indsname=M;
	ThisIsM=M;
	Where (Estimate ne 0) and (substr(Parameter,1,5) ne "State") and (substr(Parameter,1,4)
ne "Year");


	length Star $3;
	if Probt le 0.01 then Star="***";
		else if Probt le 0.05 then Star="**";
		else if Probt le 0.10 then Star="*";
		else Star="";


	if M="WORK.PEFORMODEL1" then Model="Model1";
		else if M="WORK.PEFORMODEL2" then Model="Model2";
		else if M="WORK.PEFORMODEL3" then Model="Model3";
		else if M="WORK.PEFORMODEL4" then Model="Model4";
		else if M="WORK.PEFORMODEL5" then Model="Model5";
		else if M="WORK.PEFORMODEL6" then Model="Model6";
		else if M="WORK.PEFORMODEL7" then Model="Model7";
		else if M="WORK.PEFORMODEL8" then Model="Model8";
```

```
        Results=Estimate;
        EditedResults=Cats(put(Results,comma16.4),Star);
        output;

        Results=StdErr;
        EditedResults=Cats("(",put(Results,comma16.4),")");
        output;

        keep Model Parameter EditedResults /*Probt Star*/;

run;

proc sort data=Table_Long out=Table_Long_Sorted;
        by Model Parameter;
run;

data _null_;
   put "STARTING CLEAN SPLIT OF MODEL RESULTS";
run;

data
   Model1Results(rename=(EditedResults=Model1))
   Model2Results(rename=(EditedResults=Model2))
   Model3Results(rename=(EditedResults=Model3))
   Model4Results(rename=(EditedResults=Model4))
   Model5Results(rename=(EditedResults=Model5))
   Model6Results(rename=(EditedResults=Model6))
```

```
    Model7Results(rename=(EditedResults=Model7))
    Model8Results(rename=(EditedResults=Model8));

    set Table_Long_Sorted;

    select (Model);
        when ("Model1") output Model1Results;
        when ("Model2") output Model2Results;
        when ("Model3") output Model3Results;
        when ("Model4") output Model4Results;
        when ("Model5") output Model5Results;
        when ("Model6") output Model6Results;
        when ("Model7") output Model7Results;
        when ("Model8") output Model8Results;
        otherwise;
    end;

    drop Model;
run;

/*Ordering*/
data Table_Wide;
        merge Model1Results Model2Results Model3Results Model4Results Model5Results
Model6Results Model7Results Model8Results;
        by Parameter;

        length Order 3;
```

```
        if Parameter="DID" then order=1;
                else if Parameter="State" then order=2;
                else if Parameter="Year" then order=3;
                else if Parameter="Intercept" then order=4;

        if mod(_n_,2)=1 then Regressors=Parameter;

run;

proc sort data=Table_Wide out=Table_Wide_Sorted(drop=Order Parameter);
        by Order;
run;

/*Rows for other statistics and info*/
/*Number of Obs.*/

/* Filter and format number of observations for each model */
data Obs1; set ObsModel1; if Label1="Number of Observations"; Model1 = put(NValue1,
comma16.); keep Label1 Model1; run;
data Obs2; set ObsModel2; if Label1="Number of Observations"; Model2 = put(NValue1,
comma16.); keep Label1 Model2; run;
data Obs3; set ObsModel3; if Label1="Number of Observations"; Model3 = put(NValue1,
comma16.); keep Label1 Model3; run;
data Obs4; set ObsModel4; if Label1="Number of Observations"; Model4 = put(NValue1,
comma16.); keep Label1 Model4; run;
data Obs5; set ObsModel5; if Label1="Number of Observations"; Model5 = put(NValue1,
comma16.); keep Label1 Model5; run;
data Obs6; set ObsModel6; if Label1="Number of Observations"; Model6 = put(NValue1,
comma16.); keep Label1 Model6; run;
```

```
data Obs7; set ObsModel7; if Label1="Number of Observations"; Model7 = put(NValue1, comma16.); keep Label1 Model7; run;

data Obs8; set ObsModel8; if Label1="Number of Observations"; Model8 = put(NValue1, comma16.); keep Label1 Model8; run;


/* Merge all */

data NumofObs;

	merge Obs1 Obs2 Obs3 Obs4 Obs5 Obs6 Obs7 Obs8;

	by Label1;

run;


/*Adjusted R-Square*/


/* Filter and rename each Adjusted R-Square dataset individually */

data AdjRsq1; set AdjRsqModel1 (rename=(CValue1=Model1) drop=NValue1); if Label1="Adjusted R-Square"; run;

data AdjRsq2; set AdjRsqModel2 (rename=(CValue1=Model2) drop=NValue1); if Label1="Adjusted R-Square"; run;

data AdjRsq3; set AdjRsqModel3 (rename=(CValue1=Model3) drop=NValue1); if Label1="Adjusted R-Square"; run;

data AdjRsq4; set AdjRsqModel4 (rename=(CValue1=Model4) drop=NValue1); if Label1="Adjusted R-Square"; run;

data AdjRsq5; set AdjRsqModel5 (rename=(CValue1=Model5) drop=NValue1); if Label1="Adjusted R-Square"; run;

data AdjRsq6; set AdjRsqModel6 (rename=(CValue1=Model6) drop=NValue1); if Label1="Adjusted R-Square"; run;

data AdjRsq7; set AdjRsqModel7 (rename=(CValue1=Model7) drop=NValue1); if Label1="Adjusted R-Square"; run;

data AdjRsq8; set AdjRsqModel8 (rename=(CValue1=Model8) drop=NValue1); if Label1="Adjusted R-Square"; run;
```

```
/* Merge all together */
data AdjRsq;
	merge AdjRsq1 AdjRsq2 AdjRsq3 AdjRsq4 AdjRsq5 AdjRsq6 AdjRsq7 AdjRsq8;
	by Label1;
run;

/*Overall Significance (F-Test)*/

/* Step 1: Filter and format each model's F-test result */
data OSM1; set OverallSigModel1; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
   Model1 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model1;
run;

data OSM2; set OverallSigModel2; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
   Model2 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model2;
run;

data OSM3; set OverallSigModel3; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
   Model3 = cats(put(Fvalue, comma20.2), Star);
```

```
   keep Label1 Model3;
run;

data OSM4; set OverallSigModel4; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model4 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model4;
run;

data OSM5; set OverallSigModel5; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model5 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model5;
run;

data OSM6; set OverallSigModel6; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model6 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model6;
run;

data OSM7; set OverallSigModel7; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model7 = cats(put(Fvalue, comma20.2), Star);
```

```
   keep Label1 Model7;
run;

data OSM8; set OverallSigModel8; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model8 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model8;
run;

/* Step 2: Merge all formatted outputs */
data OverallSig;
   merge OSM1 OSM2 OSM3 OSM4 OSM5 OSM6 OSM7 OSM8;
   by Label1;
run;

/*Combine F-values*/
Data OverallSig;
        merge OSM1 OSM2 OSM3 OSM4 OSM5 OSM6 OSM7 OSM8;
run;

/*Controls*/
Data Controls;
        length Label1 $30;

        Label1="Controls for State?";
        Model1="Yes"; Model2="Yes"; Model3="Yes"; Model4="Yes";
```

```
        Model5="Yes"; Model6="Yes"; Model7="Yes"; Model8="Yes";
        output;

        Label1="Controls for Year?";
        output;
run;

/*Combine Rows*/
Data OtherStat;
        set NumofObs AdjRsq OverallSig Controls;
        rename Label1=Regressors;
run;

/*Add Other Statistics to Results Table*/

Data Table_wide_sorted_withstat;
        set Table_Wide_Sorted OtherStat;
run;

/* Print Cleaned Table*/
ods excel file="/home/u63724901/MySAS/AllModelsCleanedHRP_Edu.xlsx"
options(Embedded_Titles="ON" Embedded_Footnotes="ON");
Title "Title";
footnote1 justify=left "Sources: Source";
footnote2 justify=left "Notes: robust standard errors are in parentheses. *, **, and ***
indicate 10%, 5%, and 1% significance levels, respectively.";
proc print data=Table_Wide_Sorted_withstat noobs;
```

```
        var regressors;
        var Model1-Model8 / Style(header)={just=center} style(data)={just=center};
        format Regressors $VariableName.;
run;
ods excel close;

/*//////////////Parallel Trends/////////////////*/
/* STEP 1: Merge EduPop with Implemented */
data CombineEduPop;
   merge EduPop3 Implemented;
   by State;

   /* MarketFac & EconNexus DIDs */
   if MarketFacImplemented="-" then MarketFac = 0;
   else if Year >= MarketFacImplemented then MarketFac = 1;
   else MarketFac = 0;

   if EconNexusImplemented="-" then EconNexus = 0;
   else if Year >= EconNexusImplemented then EconNexus = 1;
   else EconNexus = 0;
run;

/* STEP 2: Assign treatment and relative time */
data CombineEduPop2;
   set CombineEduPop;

   /* Control group: states without sales tax */
```

```
    if State in (2, 8, 26, 29, 37) then treatment = 0;
    else treatment = 1;

    /* Relative time from 2018 */
    rt = Year - 2018;
run;

/* STEP 3: Run parallel trends test (pre-treatment only) */
ods output ParameterEstimates=PE_ParallelTrends_EduPop
           FitStatistics=Fit_ParallelTrends_EduPop;

proc surveyreg data=CombineEduPop2;
    where Year < 2018;
    class State;
    model EduPop = rt rt*rt rt*rt*rt
                   treatment treatment*rt treatment*rt*rt treatment*rt*rt*rt
                   State
                   / solution adjrsq;
run;

ods output close;

/* STEP 4: Export to Excel */
ods excel file="/home/u63724901/MySAS/ParallelTrends_EduPop.xlsx";

ods excel options(sheet_name="Parameter Estimates");
proc print data=PE_ParallelTrends_EduPop noobs label;
```

```
   title "Parallel Trends Test – Per Capita Education Spending";
run;

ods excel options(sheet_name="Fit Statistics");
proc print data=Fit_ParallelTrends_EduPop noobs label;
   title "Fit Statistics – Per Capita Education Spending";
run;

ods excel close;

/*/////////////////Summary Stats////////////*/
ods excel file="/home/u63724901/MySAS/EduPop_Summary_All.xlsx"
options(sheet_interval='none');

/* Sheet 1: Overall Summary */
ods excel options(sheet_name='Overall Summary');
proc means data=EduPop3 n mean median std min max;
   var EduPop;
run;

/* Sheet 2: By State */
ods excel options(sheet_name='By State');
proc means data=EduPop3 n mean median std min max;
   class State;
   var EduPop;
run;
```

```
/* Sheet 3: By Year */
ods excel options(sheet_name='By Year');
proc means data=EduPop3 n mean median std min max;
   class Year;
   var EduPop;
run;

ods excel close;
/*/////Infrastructure/////*/

/*First Data*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Infra
        dbms=xlsx
        replace;
sheet="Infrastructure";
getnames=yes;
run;

Proc Sort Data=Infra;
        by State;
run;

Proc Transpose Data=Infra Out=Infra2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
```

```
"2020"n "2021"n "2022"n;
Run;

Data Infra3;
	Set Infra2;
	Year=input(_Name_,4.);
	Infra=Col1;
	keep State Year Infra;
Run;

/*Second Data*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Implemented
	dbms=xlsx
	replace;
sheet="Implemented";
getnames=yes;
run;

Proc Sort Data=Implemented;
	by State;
run;

/*Combine*/
Data Combine;
	Merge Infra3 Implemented;
	By State;
```

```
	/*MarketFac DID*/
	If MarketFacImplemented="-"  then MarketFac = 0;
	Else If Year>=MarketFacImplemented then MarketFac = 1;
	Else MarketFac=0;

	/*EconNexus DID*/
	If EconNexusImplemented="-" then EconNexus = 0;
	Else If Year>=EconNexusImplemented then EconNexus = 1;
	Else EconNexus=0;

	Keep State Year Infra MarketFac EconNexus;
Run;

/*Model 1 - Infra w/o Control Variables*/
ods output ParameterEstimates=PEforModel1 DataSummary=ObsModel1
		FitStatistics=AdjRsqModel1 Effects=OverallSigModel1;
proc surveyreg data=Combine plots=none;
  class State Year / ref=first;
  cluster State;
  model Infra = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Model 2 - LogInfra w/o Control Variables*/
data LogCombine;
	set Combine;
```

```
        LogInfra = Log(Infra);
Run;

ods output ParameterEstimates=PEforModel2 DataSummary=ObsModel2
                FitStatistics=AdjRsqModel2 Effects=OverallSigModel2;
proc surveyreg data=LogCombine plots=none;
  class State Year / ref=first;
  cluster State;
  model LogInfra = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Control Variables Uploading*/

/*Third Data - Party Control*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.PartyControl
        dbms=xlsx
        replace;
sheet="PartyControl";
getnames=yes;
run;

Proc Sort Data=PartyControl;
        by State;
run;
```

```
Proc Transpose Data=PartyControl Out=PartyControl2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PartyControl3;
	Set PartyControl2;
	Year=input(_Name_,4.);
	PartyControl=Col1;
	keep State Year PartyControl;
Run;

/*Fourth Data - GDP*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.GDP
	dbms=xlsx
	replace;
sheet="GDP";
getnames=yes;
run;

Proc Sort Data=GDP;
	by State;
run;
```

```
Proc Transpose Data=GDP Out=GDP2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data GDP3;
	Set GDP2;
	Year=input(_Name_,4.);
	GDP=Col1;
	keep State Year GDP;
Run;

/*Fifth Data - Personal Income*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.PersonalIncome
	dbms=xlsx
	replace;
sheet="PersonalIncome";
getnames=yes;
run;

Proc Sort Data=PersonalIncome;
	by State;
run;
```

```
Proc Transpose Data=PersonalIncome Out=PersonalIncome2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PersonalIncome3;
	Set PersonalIncome2;
	Year=input(_Name_,4.);
	PersonalIncome=Col1;
	keep State Year PersonalIncome;
Run;

/*Sixth Data - Population*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Pop
	dbms=xlsx
	replace;
sheet="Pop";
getnames=yes;
run;

Proc Sort Data=Pop;
	by State;
run;
```

```
Proc Transpose Data=Pop Out=Pop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data Pop3;
	Set Pop2;
	Year=input(_Name_,4.);
	Pop=Col1;
	keep State Year Pop;
Run;

/*Seventh Data - Infra/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.InfraPop
	dbms=xlsx
	replace;
sheet="InfrastructurePop";
getnames=yes;
run;

Proc Sort Data=InfraPop;
	by State;
run;
```

```
Proc Transpose Data=InfraPop Out=InfraPop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data InfraPop3;
	Set InfraPop2;
	Year=input(_Name_,4.);
	InfraPop=Col1;
	keep State Year InfraPop;
Run;

/*Eighth Data - GDP/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.GDPPop
	dbms=xlsx
	replace;
sheet="GDPPop";
getnames=yes;
run;

Proc Sort Data=GDPPop;
	by State;
run;
```

```
Proc Transpose Data=GDPPop Out=GDPPop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data GDPPop3;
	Set GDPPop2;
	Year=input(_Name_,4.);
	GDPPop=Col1;
	keep State Year GDPPop;
Run;

/*Ninth Data - Personal Income/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.PersonalIncomePop
	dbms=xlsx
	replace;
sheet="PersIncPop";
getnames=yes;
run;

Proc Sort Data=PersonalIncomePop;
	by State;
run;
```

```
Proc Transpose Data=PersonalIncomePop Out=PersonalIncomePop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PersonalIncomePop3;
	Set PersonalIncomePop2;
	Year=input(_Name_,4.);
	PersonalIncomePop=Col1;
	keep State Year PersonalIncomePop;
Run;

/*Tenth Data - Tax Rates*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Rate
	dbms=xlsx
	replace;
sheet="Rate";
getnames=yes;
run;

Proc Sort Data=Rate;
	by State;
run;
```

```
Proc Transpose Data=Rate Out=Rate2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;

Data Rate3;
	Set Rate2;
	Year=input(_Name_,4.);
	Rate=Col1;
	keep State Year Rate;
Run;

/*Eleventh Data - EN Threshold*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Threshold
	dbms=xlsx
	replace;
sheet="Threshold";
getnames=yes;
run;

Proc Sort Data=Threshold;
	by State;
run;
```

```
Proc Transpose Data=Threshold Out=Threshold2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;

Data Threshold3;
	Set Threshold2;
	Year=input(_Name_,4.);
	Threshold=Col1;
	keep State Year Threshold;
Run;

/*NEW MODELS*/

/*Combining Everything*/

/* Step 1: Merge Infra3 with Implemented (already done) */
Data Combine;
	Merge Infra3 Implemented;
	By State;

	If MarketFacImplemented="-"  then MarketFac = 0;
	Else If Year>=MarketFacImplemented then MarketFac = 1;
	Else MarketFac=0;
```

```
        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;

        keep State Year Infra MarketFac EconNexus;
Run;

/* Step 2: Merge Combine with control variables (must sort by State Year first) */
proc sort data=Combine; by State Year; run;
proc sort data=PartyControl3; by State Year; run;
proc sort data=GDP3; by State Year; run;
proc sort data=PersonalIncome3; by State Year; run;
proc sort data=Pop3; by State Year; run;
proc sort data=Rate3; by State Year; run;
proc sort data=Threshold3; by State Year; run;

/* Step 3: Sequential merge */
data CombineFull;
        merge Combine
            PartyControl3
            GDP3
            PersonalIncome3
            Pop3
            Rate3
            Threshold3;
        by State Year;
run;
```

```
/*Model 3 - Infra w/ Control Variables*/
ods output ParameterEstimates=PEforModel3 DataSummary=ObsModel3
                FitStatistics=AdjRsqModel3 Effects=OverallSigModel3;
proc surveyreg data=CombineFull plots=none;
  class State Year / ref=first;
  cluster State;
  model Infra = MarketFac EconNexus State Year PartyControl GDP PersonalIncome Pop Rate
Threshold / solution adjrsq;
run;
quit;

/*Model 4 - LogInfra w/ Control Variables*/
data LogCombineFull;
        set CombineFull;
        LogInfra = Log(Infra);
Run;

ods output ParameterEstimates=PEforModel4 DataSummary=ObsModel4
                FitStatistics=AdjRsqModel4 Effects=OverallSigModel4;
proc surveyreg data=LogCombineFull plots=none;
  class State Year / ref=first;
  cluster State;
  model LogInfra = MarketFac EconNexus State Year  PartyControl GDP PersonalIncome Pop
Rate Threshold / solution adjrsq;
run;
quit;
```

```
/*PER-CAPITA MODELS*/

/*Combine*/
Data CombinePop;
	Merge InfraPop3 Implemented;
	By State;

	/*MarketFac DID*/
	If MarketFacImplemented="-"  then MarketFac = 0;
	Else If Year>=MarketFacImplemented then MarketFac = 1;
	Else MarketFac=0;

	/*EconNexus DID*/
	If EconNexusImplemented="-" then EconNexus = 0;
	Else If Year>=EconNexusImplemented then EconNexus = 1;
	Else EconNexus=0;

	Keep State Year InfraPop MarketFac EconNexus;
Run;

/*Model 5 - InfraPop w/o Control Variables*/
ods output ParameterEstimates=PEforModel5 DataSummary=ObsModel5
		FitStatistics=AdjRsqModel5 Effects=OverallSigModel5;
proc surveyreg data=CombinePop plots=none;
  class State Year / ref=first;
  cluster State;
  model InfraPop = MarketFac EconNexus State Year / solution adjrsq;
```

```
run;
quit;

/*Model 6 - LogInfraPop w/o Control Variables*/
data LogCombinePop;
        set CombinePop;
        LogInfraPop = Log(InfraPop);
Run;

ods output ParameterEstimates=PEforModel6 DataSummary=ObsModel6
                FitStatistics=AdjRsqModel6 Effects=OverallSigModel6;
proc surveyreg data=LogCombinePop plots=none;
  class State Year / ref=first;
  cluster State;
  model LogInfraPop = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Control Variables*/
Data CombinePop2;
        Merge InfraPop3 Implemented;
        By State;

        If MarketFacImplemented="-"  then MarketFac = 0;
        Else If Year>=MarketFacImplemented then MarketFac = 1;
        Else MarketFac=0;
```

```
        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;

        keep State Year InfraPop MarketFac EconNexus PartyControl;
Run;

/* Step 2: Merge Combine with control variables (must sort by State Year first) */
proc sort data=CombinePop2; by State Year; run;
proc sort data=PartyControl3; by State Year; run;
proc sort data=GDPPop3; by State Year; run;
proc sort data=PersonalIncomePop3; by State Year; run;
proc sort data=Rate3; by State Year; run;
proc sort data=Threshold3; by State Year; run;

/* Step 3: Sequential merge */
data CombineFullPop;
        merge CombinePop2
            PartyControl3
            GDPPop3
            PersonalIncomePop3
            Rate3
            Threshold3;
        by State Year;
run;

/*Model 7 - InfraPop w/ Control Variables*/
```

```
ods output ParameterEstimates=PEforModel7 DataSummary=ObsModel7
                FitStatistics=AdjRsqModel7 Effects=OverallSigModel7;
proc surveyreg data=CombineFullPop plots=none;
  class State Year / ref=first;
  cluster State;
  model InfraPop = MarketFac EconNexus State Year PartyControl GDPPop
PersonalIncomePop Rate Threshold / solution adjrsq;
run;
quit;

/*Model 8 - LogInfraPop w/ Control Variables*/
data LogCombineFullPop;
        set CombineFullPop;
        LogInfraPop = Log(InfraPop);
Run;

ods output ParameterEstimates=PEforModel8 DataSummary=ObsModel8
                FitStatistics=AdjRsqModel8 Effects=OverallSigModel8;
proc surveyreg data=LogCombineFullPop plots=none;
  class State Year / ref=first;
  cluster State;
  model LogInfraPop = MarketFac EconNexus State Year PartyControl GDPPop
PersonalIncomePop Rate Threshold / solution adjrsq;
run;
quit;

/*Cleaning*/
Data Table_Long;
```

```
length Model $10;
length Parameter $30;
set PEforModel1 PEforModel2 PEforModel3 PEforModel4 PEforModel5 PEforModel6
PEforModel7 PEforModel8 indsname=M;
ThisIsM=M;

length Star $3;
if Probt le 0.01 then Star="***";
    else if Probt le 0.05 then Star="**";
    else if Probt le 0.10 then Star="*";
    else Star="";

if M="WORK.PEFORMODEL1" then Model="Model1";
    else if M="WORK.PEFORMODEL2" then Model="Model2";
    else if M="WORK.PEFORMODEL3" then Model="Model3";
    else if M="WORK.PEFORMODEL4" then Model="Model4";
    else if M="WORK.PEFORMODEL5" then Model="Model5";
    else if M="WORK.PEFORMODEL6" then Model="Model6";
    else if M="WORK.PEFORMODEL7" then Model="Model7";
    else if M="WORK.PEFORMODEL8" then Model="Model8";

Results=Estimate;
EditedResults=Cats(put(Results,comma16.2),Star);
output;

Results=StdErr;
EditedResults=Cats("(",put(Results,comma16.2),")");
```

```
        output;

        keep Model Parameter EditedResults /*Probt Star*/;

run;

proc sort data=Table_Long out=Table_Long_Sorted;
        by Model Parameter;
run;

/* Split cleaned estimates into individual model datasets */
data Model1Results(rename=(EditedResults=Model1));
   set Table_Long_Sorted;
   if Model = "Model1";
   drop Model;
run;

data Model2Results(rename=(EditedResults=Model2));
   set Table_Long_Sorted;
   if Model = "Model2";
   drop Model;
run;

data Model3Results(rename=(EditedResults=Model3));
   set Table_Long_Sorted;
   if Model = "Model3";
   drop Model;
```

```
run;

data Model4Results(rename=(EditedResults=Model4));
    set Table_Long_Sorted;
    if Model = "Model4";
    drop Model;
run;

data Model5Results(rename=(EditedResults=Model5));
    set Table_Long_Sorted;
    if Model = "Model5";
    drop Model;
run;

data Model6Results(rename=(EditedResults=Model6));
    set Table_Long_Sorted;
    if Model = "Model6";
    drop Model;
run;

data Model7Results(rename=(EditedResults=Model7));
    set Table_Long_Sorted;
    if Model = "Model7";
    drop Model;
run;

data Model8Results(rename=(EditedResults=Model8));
```

```
    set Table_Long_Sorted;
    if Model = "Model8";
    drop Model;
run;

data Table_Wide;
        merge Model1Results Model2Results Model3Results Model4Results Model5Results
Model6Results Model7Results Model8Results;
        by Parameter;

        length Order 3;
        if Parameter="week" then Order=1;
                else if Parameter="age" then Order=2;
                else Order=3;

        if mod(_n_,2)=1 then Regressors=Parameter;

run;

proc sort data=Table_Wide out=Table_Wide_Sorted(drop=Order Parameter);
        by Order;
run;

Data Table_Long;
        length Model $10;
        length Parameter $30;
        set PEforModel1 PEforModel2 PEforModel3 PEforModel4 PEforModel5 PEforModel6
PEforModel7 PEforModel8 indsname=M;
```

```
ThisIsM=M;
Where (Estimate ne 0) and (substr(Parameter,1,5) ne "State") and (substr(Parameter,1,4)
ne "Year");

length Star $3;
if Probt le 0.01 then Star="***";
    else if Probt le 0.05 then Star="**";
    else if Probt le 0.10 then Star="*";
    else Star="";

if M="WORK.PEFORMODEL1" then Model="Model1";
    else if M="WORK.PEFORMODEL2" then Model="Model2";
    else if M="WORK.PEFORMODEL3" then Model="Model3";
    else if M="WORK.PEFORMODEL4" then Model="Model4";
    else if M="WORK.PEFORMODEL5" then Model="Model5";
    else if M="WORK.PEFORMODEL6" then Model="Model6";
    else if M="WORK.PEFORMODEL7" then Model="Model7";
    else if M="WORK.PEFORMODEL8" then Model="Model8";

Results=Estimate;
EditedResults=Cats(put(Results,comma16.4),Star);
output;

Results=StdErr;
EditedResults=Cats("(",put(Results,comma16.4),")");
output;
```

```
        keep Model Parameter EditedResults /*Probt Star*/;

run;

proc sort data=Table_Long out=Table_Long_Sorted;
        by Model Parameter;
run;

data _null_;
   put "STARTING CLEAN SPLIT OF MODEL RESULTS";
run;

data
   Model1Results(rename=(EditedResults=Model1))
   Model2Results(rename=(EditedResults=Model2))
   Model3Results(rename=(EditedResults=Model3))
   Model4Results(rename=(EditedResults=Model4))
   Model5Results(rename=(EditedResults=Model5))
   Model6Results(rename=(EditedResults=Model6))
   Model7Results(rename=(EditedResults=Model7))
   Model8Results(rename=(EditedResults=Model8));

   set Table_Long_Sorted;

   select (Model);
      when ("Model1") output Model1Results;
      when ("Model2") output Model2Results;
```

```
        when ("Model3") output Model3Results;
        when ("Model4") output Model4Results;
        when ("Model5") output Model5Results;
        when ("Model6") output Model6Results;
        when ("Model7") output Model7Results;
        when ("Model8") output Model8Results;
        otherwise;
    end;

    drop Model;
run;

/*Ordering*/
data Table_Wide;
        merge Model1Results Model2Results Model3Results Model4Results Model5Results
Model6Results Model7Results Model8Results;
        by Parameter;

        length Order 3;
        if Parameter="DID" then order=1;
                else if Parameter="State" then order=2;
                else if Parameter="Year" then order=3;
                else if Parameter="Intercept" then order=4;

        if mod(_n_,2)=1 then Regressors=Parameter;

run;
```

```
proc sort data=Table_Wide out=Table_Wide_Sorted(drop=Order Parameter);
	by Order;
run;

/*Rows for other statistics and info*/
/*Number of Obs.*/

/* Filter and format number of observations for each model */
data Obs1; set ObsModel1; if Label1="Number of Observations"; Model1 = put(NValue1, comma16.); keep Label1 Model1; run;
data Obs2; set ObsModel2; if Label1="Number of Observations"; Model2 = put(NValue1, comma16.); keep Label1 Model2; run;
data Obs3; set ObsModel3; if Label1="Number of Observations"; Model3 = put(NValue1, comma16.); keep Label1 Model3; run;
data Obs4; set ObsModel4; if Label1="Number of Observations"; Model4 = put(NValue1, comma16.); keep Label1 Model4; run;
data Obs5; set ObsModel5; if Label1="Number of Observations"; Model5 = put(NValue1, comma16.); keep Label1 Model5; run;
data Obs6; set ObsModel6; if Label1="Number of Observations"; Model6 = put(NValue1, comma16.); keep Label1 Model6; run;
data Obs7; set ObsModel7; if Label1="Number of Observations"; Model7 = put(NValue1, comma16.); keep Label1 Model7; run;
data Obs8; set ObsModel8; if Label1="Number of Observations"; Model8 = put(NValue1, comma16.); keep Label1 Model8; run;

/* Merge all */
data NumofObs;
	merge Obs1 Obs2 Obs3 Obs4 Obs5 Obs6 Obs7 Obs8;
	by Label1;
```

```
run;

/*Adjusted R-Square*/

/* Filter and rename each Adjusted R-Square dataset individually */
data AdjRsq1; set AdjRsqModel1 (rename=(CValue1=Model1) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq2; set AdjRsqModel2 (rename=(CValue1=Model2) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq3; set AdjRsqModel3 (rename=(CValue1=Model3) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq4; set AdjRsqModel4 (rename=(CValue1=Model4) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq5; set AdjRsqModel5 (rename=(CValue1=Model5) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq6; set AdjRsqModel6 (rename=(CValue1=Model6) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq7; set AdjRsqModel7 (rename=(CValue1=Model7) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq8; set AdjRsqModel8 (rename=(CValue1=Model8) drop=NValue1); if
Label1="Adjusted R-Square"; run;

/* Merge all together */
data AdjRsq;
        merge AdjRsq1 AdjRsq2 AdjRsq3 AdjRsq4 AdjRsq5 AdjRsq6 AdjRsq7 AdjRsq8;
        by Label1;
run;

/*Overall Significance (F-Test)*/
```

```
/* Step 1: Filter and format each model's F-test result */
data OSM1; set OverallSigModel1; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model1 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model1;
run;

data OSM2; set OverallSigModel2; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model2 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model2;
run;

data OSM3; set OverallSigModel3; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model3 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model3;
run;

data OSM4; set OverallSigModel4; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model4 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model4;
run;
```

```
data OSM5; set OverallSigModel5; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
   Model5 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model5;
run;

data OSM6; set OverallSigModel6; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
   Model6 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model6;
run;

data OSM7; set OverallSigModel7; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
   Model7 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model7;
run;

data OSM8; set OverallSigModel8; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";
   Model8 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model8;
run;
```

```
/* Step 2: Merge all formatted outputs */
data OverallSig;
   merge OSM1 OSM2 OSM3 OSM4 OSM5 OSM6 OSM7 OSM8;
   by Label1;
run;

/*Combine F-values*/
Data OverallSig;
        merge OSM1 OSM2 OSM3 OSM4 OSM5 OSM6 OSM7 OSM8;
run;

/*Controls*/
Data Controls;
        length Label1 $30;

        Label1="Controls for State?";
        Model1="Yes"; Model2="Yes"; Model3="Yes"; Model4="Yes";
        Model5="Yes"; Model6="Yes"; Model7="Yes"; Model8="Yes";
        output;

        Label1="Controls for Year?";
        output;
run;

/*Combine Rows*/
Data OtherStat;
```

```
        set NumofObs AdjRsq OverallSig Controls;
        rename Label1=Regressors;
run;

/*Add Other Statistics to Results Table*/

Data Table_wide_sorted_withstat;
        set Table_Wide_Sorted OtherStat;
run;

/* Print Cleaned Table*/
ods excel file="/home/u63724901/MySAS/AllModelsCleanedHRP_Infra.xlsx"
options(Embedded_Titles="ON" Embedded_Footnotes="ON");
Title "Title";
footnote1 justify=left "Sources: Source";
footnote2 justify=left "Notes: robust standard errors are in parentheses. *, **, and ***
indicate 10%, 5%, and 1% significance levels, respectively.";
proc print data=Table_Wide_Sorted_withstat noobs;
        var regressors;
        var Model1-Model8 / Style(header)={just=center} style(data)={just=center};
        format Regressors $VariableName.;
run;
ods excel close;
```

```
/*//////////////Parallel Trends/////////////////*/
/* STEP 1: Merge InfraPop with Implemented */
data CombineInfraPop;
   merge InfraPop3 Implemented;
   by State;

   /* MarketFac & EconNexus DIDs */
   if MarketFacImplemented="-" then MarketFac = 0;
   else if Year >= MarketFacImplemented then MarketFac = 1;
   else MarketFac = 0;

   if EconNexusImplemented="-" then EconNexus = 0;
   else if Year >= EconNexusImplemented then EconNexus = 1;
   else EconNexus = 0;
run;

/* STEP 2: Assign treatment and relative time */
data CombineInfraPop2;
   set CombineInfraPop;
```

```
    /* Control group: states without sales tax */
    if State in (2, 8, 26, 29, 37) then treatment = 0;
    else treatment = 1;

    /* Relative time from 2018 */
    rt = Year - 2018;
run;

/* STEP 3: Run parallel trends test (pre-treatment only) */
ods output ParameterEstimates=PE_ParallelTrends_InfraPop
           FitStatistics=Fit_ParallelTrends_InfraPop;

proc surveyreg data=CombineInfraPop2;
    where Year < 2018;
    class State;
    model InfraPop = rt rt*rt rt*rt*rt
                     treatment treatment*rt treatment*rt*rt treatment*rt*rt*rt
                     State
                     / solution adjrsq;
run;

ods output close;

/* STEP 4: Export to Excel */
ods excel file="/home/u63724901/MySAS/ParallelTrends_InfraPop.xlsx";

ods excel options(sheet_name="Parameter Estimates");
```

```
proc print data=PE_ParallelTrends_InfraPop noobs label;
   title "Parallel Trends Test – Per Capita Infra";
run;

ods excel options(sheet_name="Fit Statistics");
proc print data=Fit_ParallelTrends_InfraPop noobs label;
   title "Fit Statistics – Per Capita Infra";
run;

ods excel close;

/*Summary Stats*/
ods excel file="/home/u63724901/MySAS/InfraPop_Summary_All.xlsx"
options(sheet_interval='none');

/* Sheet 1: Overall Summary */
ods excel options(sheet_name='Overall Summary');
proc means data=InfraPop3 n mean median std min max;
   var InfraPop;
run;

/* Sheet 2: By State */
ods excel options(sheet_name='By State');
proc means data=InfraPop3 n mean median std min max;
   class State;
   var InfraPop;
run;
```

```
/* Sheet 3: By Year */
ods excel options(sheet_name='By Year');
proc means data=InfraPop3 n mean median std min max;
   class Year;
   var InfraPop;
run;

ods excel close;
/*/////Security/////*/

/*First Data*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Security
        dbms=xlsx
        replace;
sheet="Security";
getnames=yes;
run;

Proc Sort Data=Security;
        by State;
run;

Proc Transpose Data=Security Out=Security2;
        by State;
```

```
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data Security3;
        Set Security2;
        Year=input(_Name_,4.);
        Security=Col1;
        keep State Year Security;
Run;

/*Second Data*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Implemented
        dbms=xlsx
        replace;
sheet="Implemented";
getnames=yes;
run;

Proc Sort Data=Implemented;
        by State;
run;

/*Combine*/
Data Combine;
```

```
        Merge Security3 Implemented;
        By State;

        /*MarketFac DID*/
        If MarketFacImplemented="-"  then MarketFac = 0;
        Else If Year>=MarketFacImplemented then MarketFac = 1;
        Else MarketFac=0;

        /*EconNexus DID*/
        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;

        Keep State Year Security MarketFac EconNexus;
Run;

/*Model 1 - Security w/o Control Variables*/
ods output ParameterEstimates=PEforModel1 DataSummary=ObsModel1
                FitStatistics=AdjRsqModel1 Effects=OverallSigModel1;
proc surveyreg data=Combine plots=none;
  class State Year / ref=first;
  cluster State;
  model Security = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Model 2 - LogTaxRev w/o Control Variables*/
```

```
data LogCombine;
        set Combine;
        LogSecurity = Log(Security);
Run;

ods output ParameterEstimates=PEforModel2 DataSummary=ObsModel2
                FitStatistics=AdjRsqModel2 Effects=OverallSigModel2;
proc surveyreg data=LogCombine plots=none;
  class State Year / ref=first;
  cluster State;
  model LogSecurity = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Control Variables Uploading*/

/*Third Data - Party Control*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.PartyControl
        dbms=xlsx
        replace;
sheet="PartyControl";
getnames=yes;
run;

Proc Sort Data=PartyControl;
        by State;
```

```
run;

Proc Transpose Data=PartyControl Out=PartyControl2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PartyControl3;
	Set PartyControl2;
	Year=input(_Name_,4.);
	PartyControl=Col1;
	keep State Year PartyControl;
Run;

/*Fourth Data - GDP*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.GDP
	dbms=xlsx
	replace;
sheet="GDP";
getnames=yes;
run;

Proc Sort Data=GDP;
	by State;
```

```
run;

Proc Transpose Data=GDP Out=GDP2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data GDP3;
        Set GDP2;
        Year=input(_Name_,4.);
        GDP=Col1;
        keep State Year GDP;
Run;

/*Fifth Data - Personal Income*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.PersonalIncome
        dbms=xlsx
        replace;
sheet="PersonalIncome";
getnames=yes;
run;

Proc Sort Data=PersonalIncome;
        by State;
```

```
run;

Proc Transpose Data=PersonalIncome Out=PersonalIncome2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PersonalIncome3;
        Set PersonalIncome2;
        Year=input(_Name_,4.);
        PersonalIncome=Col1;
        keep State Year PersonalIncome;
Run;

/*Sixth Data - Population*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Pop
        dbms=xlsx
        replace;
sheet="Pop";
getnames=yes;
run;

Proc Sort Data=Pop;
        by State;
```

```
run;

Proc Transpose Data=Pop Out=Pop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data Pop3;
	Set Pop2;
	Year=input(_Name_,4.);
	Pop=Col1;
	keep State Year Pop;
Run;

/*Seventh Data - Security/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.SecurityPop
	dbms=xlsx
	replace;
sheet="SecurityPop";
getnames=yes;
run;

Proc Sort Data=SecurityPop;
	by State;
```

```
run;

Proc Transpose Data=SecurityPop Out=SecurityPop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data SecurityPop3;
	Set SecurityPop2;
	Year=input(_Name_,4.);
	SecurityPop=Col1;
	keep State Year SecurityPop;
Run;

/*Eighth Data - GDP/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.GDPPop
	dbms=xlsx
	replace;
sheet="GDPPop";
getnames=yes;
run;

Proc Sort Data=GDPPop;
	by State;
```

```
run;

Proc Transpose Data=GDPPop Out=GDPPop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data GDPPop3;
	Set GDPPop2;
	Year=input(_Name_,4.);
	GDPPop=Col1;
	keep State Year GDPPop;
Run;

/*Ninth Data - Personal Income/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.PersonalIncomePop
	dbms=xlsx
	replace;
sheet="PersIncPop";
getnames=yes;
run;

Proc Sort Data=PersonalIncomePop;
	by State;
```

```
run;

Proc Transpose Data=PersonalIncomePop Out=PersonalIncomePop2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PersonalIncomePop3;
	Set PersonalIncomePop2;
	Year=input(_Name_,4.);
	PersonalIncomePop=Col1;
	keep State Year PersonalIncomePop;
Run;

/*Tenth Data - Tax Rates*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Rate
	dbms=xlsx
	replace;
sheet="Rate";
getnames=yes;
run;

Proc Sort Data=Rate;
	by State;
```

```
run;

Proc Transpose Data=Rate Out=Rate2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;

Data Rate3;
	Set Rate2;
	Year=input(_Name_,4.);
	Rate=Col1;
	keep State Year Rate;
Run;

/*Eleventh Data - EN Threshold*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Threshold
	dbms=xlsx
	replace;
sheet="Threshold";
getnames=yes;
run;

Proc Sort Data=Threshold;
	by State;
```

```
run;

Proc Transpose Data=Threshold Out=Threshold2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;

Data Threshold3;
	Set Threshold2;
	Year=input(_Name_,4.);
	Threshold=Col1;
	keep State Year Threshold;
Run;

/*NEW MODELS*/

/*Combining Everything*/

/* Step 1: Merge Security3 with Implemented (already done) */
Data Combine;
	Merge Security3 Implemented;
	By State;

	If MarketFacImplemented="-"  then MarketFac = 0;
	Else If Year>=MarketFacImplemented then MarketFac = 1;
```

```
        Else MarketFac=0;

        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;

        keep State Year Security MarketFac EconNexus;
Run;

/* Step 2: Merge Combine with control variables (must sort by State Year first) */
proc sort data=Combine; by State Year; run;
proc sort data=PartyControl3; by State Year; run;
proc sort data=GDP3; by State Year; run;
proc sort data=PersonalIncome3; by State Year; run;
proc sort data=Pop3; by State Year; run;
proc sort data=Rate3; by State Year; run;
proc sort data=Threshold3; by State Year; run;

/* Step 3: Sequential merge */
data CombineFull;
        merge Combine
            PartyControl3
            GDP3
            PersonalIncome3
            Pop3
            Rate3
            Threshold3;
```

```
	by State Year;
run;

/*Model 3 - Security w/ Control Variables*/
ods output ParameterEstimates=PEforModel3 DataSummary=ObsModel3
		FitStatistics=AdjRsqModel3 Effects=OverallSigModel3;
proc surveyreg data=CombineFull plots=none;
  class State Year / ref=first;
  cluster State;
  model Security = MarketFac EconNexus State Year PartyControl GDP PersonalIncome Pop
Rate Threshold / solution adjrsq;
run;
quit;

/*Model 4 - LogSecurity w/ Control Variables*/
data LogCombineFull;
	set CombineFull;
	LogSecurity = Log(Security);
Run;

ods output ParameterEstimates=PEforModel4 DataSummary=ObsModel4
		FitStatistics=AdjRsqModel4 Effects=OverallSigModel4;
proc surveyreg data=LogCombineFull plots=none;
  class State Year / ref=first;
  cluster State;
  model LogSecurity = MarketFac EconNexus State Year  PartyControl GDP PersonalIncome
Pop Rate Threshold / solution adjrsq;
run;
```

```
quit;

/*PER-CAPITA MODELS*/

/*Combine*/
Data CombinePop;
	Merge SecurityPop3 Implemented;
	By State;

	/*MarketFac DID*/
	If MarketFacImplemented="-"  then MarketFac = 0;
	Else If Year>=MarketFacImplemented then MarketFac = 1;
	Else MarketFac=0;

	/*EconNexus DID*/
	If EconNexusImplemented="-" then EconNexus = 0;
	Else If Year>=EconNexusImplemented then EconNexus = 1;
	Else EconNexus=0;

	Keep State Year SecurityPop MarketFac EconNexus;
Run;

/*Model 5 - SecurityPop w/o Control Variables*/
ods output ParameterEstimates=PEforModel5 DataSummary=ObsModel5
		FitStatistics=AdjRsqModel5 Effects=OverallSigModel5;
proc surveyreg data=CombinePop plots=none;
  class State Year / ref=first;
```

```
  cluster State;
  model SecurityPop = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Model 6 - LogSecurityPop w/o Control Variables*/
data LogCombinePop;
        set CombinePop;
        LogSecurityPop = Log(SecurityPop);
Run;

ods output ParameterEstimates=PEforModel6 DataSummary=ObsModel6
                FitStatistics=AdjRsqModel6 Effects=OverallSigModel6;
proc surveyreg data=LogCombinePop plots=none;
  class State Year / ref=first;
  cluster State;
  model LogSecurityPop = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Control Variables*/
Data CombinePop2;
        Merge SecurityPop3 Implemented;
        By State;

        If MarketFacImplemented="-"  then MarketFac = 0;
        Else If Year>=MarketFacImplemented then MarketFac = 1;
```

```
        Else MarketFac=0;

        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;

        keep State Year SecurityPop MarketFac EconNexus PartyControl;
Run;

/* Step 2: Merge Combine with control variables (must sort by State Year first) */
proc sort data=CombinePop2; by State Year; run;
proc sort data=PartyControl3; by State Year; run;
proc sort data=GDPPop3; by State Year; run;
proc sort data=PersonalIncomePop3; by State Year; run;
proc sort data=Rate3; by State Year; run;
proc sort data=Threshold3; by State Year; run;

/* Step 3: Sequential merge */
data CombineFullPop;
        merge CombinePop2
            PartyControl3
            GDPPop3
            PersonalIncomePop3
            Rate3
            Threshold3;
        by State Year;
run;
```

```
/*Model 7 - SecurityPop w/ Control Variables*/
ods output ParameterEstimates=PEforModel7 DataSummary=ObsModel7
                FitStatistics=AdjRsqModel7 Effects=OverallSigModel7;
proc surveyreg data=CombineFullPop plots=none;
  class State Year / ref=first;
  cluster State;
  model SecurityPop = MarketFac EconNexus State Year PartyControl GDPPop
PersonalIncomePop Rate Threshold / solution adjrsq;
run;
quit;

/*Model 8 - LogSecurityPop w/ Control Variables*/
data LogCombineFullPop;
        set CombineFullPop;
        LogSecurityPop = Log(SecurityPop);
Run;

ods output ParameterEstimates=PEforModel8 DataSummary=ObsModel8
                FitStatistics=AdjRsqModel8 Effects=OverallSigModel8;
proc surveyreg data=LogCombineFullPop plots=none;
  class State Year / ref=first;
  cluster State;
  model LogSecurityPop = MarketFac EconNexus State Year PartyControl GDPPop
PersonalIncomePop Rate Threshold / solution adjrsq;
run;
quit;
```

```
/*Cleaning*/
Data Table_Long;
	length Model $10;
	length Parameter $30;
	set PEforModel1 PEforModel2 PEforModel3 PEforModel4 PEforModel5 PEforModel6
PEforModel7 PEforModel8 indsname=M;
	ThisIsM=M;

	length Star $3;
	if Probt le 0.01 then Star="***";
		else if Probt le 0.05 then Star="**";
		else if Probt le 0.10 then Star="*";
		else Star="";

	if M="WORK.PEFORMODEL1" then Model="Model1";
		else if M="WORK.PEFORMODEL2" then Model="Model2";
		else if M="WORK.PEFORMODEL3" then Model="Model3";
		else if M="WORK.PEFORMODEL4" then Model="Model4";
		else if M="WORK.PEFORMODEL5" then Model="Model5";
		else if M="WORK.PEFORMODEL6" then Model="Model6";
		else if M="WORK.PEFORMODEL7" then Model="Model7";
		else if M="WORK.PEFORMODEL8" then Model="Model8";

	Results=Estimate;
	EditedResults=Cats(put(Results,comma16.2),Star);
	output;
```

```
        Results=StdErr;
        EditedResults=Cats("(",put(Results,comma16.2),")");
        output;

        keep Model Parameter EditedResults /*Probt Star*/;

run;

proc sort data=Table_Long out=Table_Long_Sorted;
        by Model Parameter;
run;

/* Split cleaned estimates into individual model datasets */
data Model1Results(rename=(EditedResults=Model1));
   set Table_Long_Sorted;
   if Model = "Model1";
   drop Model;
run;

data Model2Results(rename=(EditedResults=Model2));
   set Table_Long_Sorted;
   if Model = "Model2";
   drop Model;
run;

data Model3Results(rename=(EditedResults=Model3));
   set Table_Long_Sorted;
```

```
    if Model = "Model3";
    drop Model;
run;

data Model4Results(rename=(EditedResults=Model4));
    set Table_Long_Sorted;
    if Model = "Model4";
    drop Model;
run;

data Model5Results(rename=(EditedResults=Model5));
    set Table_Long_Sorted;
    if Model = "Model5";
    drop Model;
run;

data Model6Results(rename=(EditedResults=Model6));
    set Table_Long_Sorted;
    if Model = "Model6";
    drop Model;
run;

data Model7Results(rename=(EditedResults=Model7));
    set Table_Long_Sorted;
    if Model = "Model7";
    drop Model;
run;
```

```
data Model8Results(rename=(EditedResults=Model8));
   set Table_Long_Sorted;
   if Model = "Model8";
   drop Model;
run;

data Table_Wide;
        merge Model1Results Model2Results Model3Results Model4Results Model5Results
Model6Results Model7Results Model8Results;
        by Parameter;

        length Order 3;
        if Parameter="week" then Order=1;
                else if Parameter="age" then Order=2;
                else Order=3;

        if mod(_n_,2)=1 then Regressors=Parameter;

run;

proc sort data=Table_Wide out=Table_Wide_Sorted(drop=Order Parameter);
        by Order;
run;

Data Table_Long;
        length Model $10;
```

```
length Parameter $30;
set PEforModel1 PEforModel2 PEforModel3 PEforModel4 PEforModel5 PEforModel6
PEforModel7 PEforModel8 indsname=M;
ThisIsM=M;
Where (Estimate ne 0) and (substr(Parameter,1,5) ne "State") and (substr(Parameter,1,4)
ne "Year");


length Star $3;
if Probt le 0.01 then Star="***";
        else if Probt le 0.05 then Star="**";
        else if Probt le 0.10 then Star="*";
        else Star="";


if M="WORK.PEFORMODEL1" then Model="Model1";
        else if M="WORK.PEFORMODEL2" then Model="Model2";
        else if M="WORK.PEFORMODEL3" then Model="Model3";
        else if M="WORK.PEFORMODEL4" then Model="Model4";
        else if M="WORK.PEFORMODEL5" then Model="Model5";
        else if M="WORK.PEFORMODEL6" then Model="Model6";
        else if M="WORK.PEFORMODEL7" then Model="Model7";
        else if M="WORK.PEFORMODEL8" then Model="Model8";


Results=Estimate;
EditedResults=Cats(put(Results,comma16.4),Star);
output;


Results=StdErr;
EditedResults=Cats("(",put(Results,comma16.4),")");
```

```
        output;

        keep Model Parameter EditedResults /*Probt Star*/;

run;

proc sort data=Table_Long out=Table_Long_Sorted;
        by Model Parameter;
run;

data _null_;
   put "STARTING CLEAN SPLIT OF MODEL RESULTS";
run;

data
   Model1Results(rename=(EditedResults=Model1))
   Model2Results(rename=(EditedResults=Model2))
   Model3Results(rename=(EditedResults=Model3))
   Model4Results(rename=(EditedResults=Model4))
   Model5Results(rename=(EditedResults=Model5))
   Model6Results(rename=(EditedResults=Model6))
   Model7Results(rename=(EditedResults=Model7))
   Model8Results(rename=(EditedResults=Model8));

   set Table_Long_Sorted;

   select (Model);
```

```
        when ("Model1") output Model1Results;
        when ("Model2") output Model2Results;
        when ("Model3") output Model3Results;
        when ("Model4") output Model4Results;
        when ("Model5") output Model5Results;
        when ("Model6") output Model6Results;
        when ("Model7") output Model7Results;
        when ("Model8") output Model8Results;
        otherwise;
    end;

    drop Model;
run;

/*Ordering*/
data Table_Wide;
        merge Model1Results Model2Results Model3Results Model4Results Model5Results
Model6Results Model7Results Model8Results;
        by Parameter;

        length Order 3;
        if Parameter="DID" then order=1;
                else if Parameter="State" then order=2;
                else if Parameter="Year" then order=3;
                else if Parameter="Intercept" then order=4;

        if mod(_n_,2)=1 then Regressors=Parameter;
```

```
run;

proc sort data=Table_Wide out=Table_Wide_Sorted(drop=Order Parameter);
        by Order;
run;

/*Rows for other statistics and info*/
/*Number of Obs.*/

/* Filter and format number of observations for each model */
data Obs1; set ObsModel1; if Label1="Number of Observations"; Model1 = put(NValue1,
comma16.); keep Label1 Model1; run;
data Obs2; set ObsModel2; if Label1="Number of Observations"; Model2 = put(NValue1,
comma16.); keep Label1 Model2; run;
data Obs3; set ObsModel3; if Label1="Number of Observations"; Model3 = put(NValue1,
comma16.); keep Label1 Model3; run;
data Obs4; set ObsModel4; if Label1="Number of Observations"; Model4 = put(NValue1,
comma16.); keep Label1 Model4; run;
data Obs5; set ObsModel5; if Label1="Number of Observations"; Model5 = put(NValue1,
comma16.); keep Label1 Model5; run;
data Obs6; set ObsModel6; if Label1="Number of Observations"; Model6 = put(NValue1,
comma16.); keep Label1 Model6; run;
data Obs7; set ObsModel7; if Label1="Number of Observations"; Model7 = put(NValue1,
comma16.); keep Label1 Model7; run;
data Obs8; set ObsModel8; if Label1="Number of Observations"; Model8 = put(NValue1,
comma16.); keep Label1 Model8; run;

/* Merge all */
data NumofObs;
```

```
	merge Obs1 Obs2 Obs3 Obs4 Obs5 Obs6 Obs7 Obs8;
	by Label1;
run;

/*Adjusted R-Square*/

/* Filter and rename each Adjusted R-Square dataset individually */
data AdjRsq1; set AdjRsqModel1 (rename=(CValue1=Model1) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq2; set AdjRsqModel2 (rename=(CValue1=Model2) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq3; set AdjRsqModel3 (rename=(CValue1=Model3) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq4; set AdjRsqModel4 (rename=(CValue1=Model4) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq5; set AdjRsqModel5 (rename=(CValue1=Model5) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq6; set AdjRsqModel6 (rename=(CValue1=Model6) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq7; set AdjRsqModel7 (rename=(CValue1=Model7) drop=NValue1); if
Label1="Adjusted R-Square"; run;
data AdjRsq8; set AdjRsqModel8 (rename=(CValue1=Model8) drop=NValue1); if
Label1="Adjusted R-Square"; run;

/* Merge all together */
data AdjRsq;
	merge AdjRsq1 AdjRsq2 AdjRsq3 AdjRsq4 AdjRsq5 AdjRsq6 AdjRsq7 AdjRsq8;
	by Label1;
run;
```

```
/*Overall Significance (F-Test)*/

/* Step 1: Filter and format each model's F-test result */
data OSM1; set OverallSigModel1; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model1 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model1;
run;

data OSM2; set OverallSigModel2; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model2 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model2;
run;

data OSM3; set OverallSigModel3; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model3 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model3;
run;

data OSM4; set OverallSigModel4; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model4 = cats(put(Fvalue, comma20.2), Star);
```

```
   keep Label1 Model4;
run;

data OSM5; set OverallSigModel5; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model5 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model5;
run;

data OSM6; set OverallSigModel6; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model6 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model6;
run;

data OSM7; set OverallSigModel7; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model7 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model7;
run;

data OSM8; set OverallSigModel8; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model8 = cats(put(Fvalue, comma20.2), Star);
```

```
   keep Label1 Model8;
run;

/* Step 2: Merge all formatted outputs */
data OverallSig;
   merge OSM1 OSM2 OSM3 OSM4 OSM5 OSM6 OSM7 OSM8;
   by Label1;
run;

/*Combine F-values*/
Data OverallSig;
        merge OSM1 OSM2 OSM3 OSM4 OSM5 OSM6 OSM7 OSM8;
run;

/*Controls*/
Data Controls;
        length Label1 $30;

        Label1="Controls for State?";
        Model1="Yes"; Model2="Yes"; Model3="Yes"; Model4="Yes";
        Model5="Yes"; Model6="Yes"; Model7="Yes"; Model8="Yes";
        output;

        Label1="Controls for Year?";
        output;
run;
```

```
/*Combine Rows*/
Data OtherStat;
	set NumofObs AdjRsq OverallSig Controls;
	rename Label1=Regressors;
run;

/*Add Other Statistics to Results Table*/

Data Table_wide_sorted_withstat;
	set Table_Wide_Sorted OtherStat;
run;

/* Print Cleaned Table*/
ods excel file="/home/u63724901/MySAS/AllModelsCleanedHRP_Security.xlsx"
options(Embedded_Titles="ON" Embedded_Footnotes="ON");
Title "Title";
footnote1 justify=left "Sources: Source";
footnote2 justify=left "Notes: robust standard errors are in parentheses. *, **, and ***
indicate 10%, 5%, and 1% significance levels, respectively.";
proc print data=Table_Wide_Sorted_withstat noobs;
	var regressors;
	var Model1-Model8 / Style(header)={just=center} style(data)={just=center};
	format Regressors $VariableName.;
run;
ods excel close;
```

```
/*//////////////Parallel Trends/////////////////*/
/* STEP 1: Merge SecurityPop with Implemented */
data CombineSecurityPop;
   merge SecurityPop3 Implemented;
   by State;

   /* MarketFac & EconNexus DIDs */
   if MarketFacImplemented="-" then MarketFac = 0;
   else if Year >= MarketFacImplemented then MarketFac = 1;
   else MarketFac = 0;
```

```
    if EconNexusImplemented="-" then EconNexus = 0;
    else if Year >= EconNexusImplemented then EconNexus = 1;
    else EconNexus = 0;
run;

/* STEP 2: Assign treatment and relative time */
data CombineSecurityPop2;
    set CombineSecurityPop;

    /* Control group: states without sales tax */
    if State in (2, 8, 26, 29, 37) then treatment = 0;
    else treatment = 1;

    /* Relative time from 2018 */
    rt = Year - 2018;
run;

/* STEP 3: Run parallel trends test (pre-treatment only) */
ods output ParameterEstimates=PE_ParallelTrends_SecurityPop
           FitStatistics=Fit_ParallelTrends_SecurityPop;

proc surveyreg data=CombineSecurityPop2;
    where Year < 2018;
    class State;
    model SecurityPop = rt rt*rt rt*rt*rt
                treatment treatment*rt treatment*rt*rt treatment*rt*rt*rt
                State
```

```
                / solution adjrsq;
run;

ods output close;

/* STEP 4: Export to Excel */
ods excel file="/home/u63724901/MySAS/ParallelTrends_SecurityPop.xlsx";

ods excel options(sheet_name="Parameter Estimates");
proc print data=PE_ParallelTrends_SecurityPop noobs label;
   title "Parallel Trends Test – Per Capita Security Spending";
run;

ods excel options(sheet_name="Fit Statistics");
proc print data=Fit_ParallelTrends_SecurityPop noobs label;
   title "Fit Statistics – Per Capita Security Spending";
run;

ods excel close;

/*Summary Stats*/
ods excel file="/home/u63724901/MySAS/SecurityPop_Summary_All.xlsx"
options(sheet_interval='none');

/* Sheet 1: Overall Summary */
ods excel options(sheet_name='Overall Summary');
proc means data=SecurityPop3 n mean median std min max;
```

```
    var SecurityPop;
run;

/* Sheet 2: By State */
ods excel options(sheet_name='By State');
proc means data=SecurityPop3 n mean median std min max;
    class State;
    var SecurityPop;
run;

/* Sheet 3: By Year */
ods excel options(sheet_name='By Year');
proc means data=SecurityPop3 n mean median std min max;
    class Year;
    var SecurityPop;
run;

ods excel close;
/*/////Social/////*/

/*First Data*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
            out=work.Social
            dbms=xlsx
            replace;
sheet="Social";
getnames=yes;
```

```
run;

Proc Sort Data=Social;
	by State;
run;

Proc Transpose Data=Social Out=Social2;
	by State;
	var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data Social3;
	Set Social2;
	Year=input(_Name_,4.);
	Social=Col1;
	keep State Year Social;
Run;

/*Second Data*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
	out=work.Implemented
	dbms=xlsx
	replace;
sheet="Implemented";
getnames=yes;
```

```
run;

Proc Sort Data=Implemented;
	by State;
run;

/*Combine*/
Data Combine;
	Merge Social3 Implemented;
	By State;

	/*MarketFac DID*/
	If MarketFacImplemented="-"  then MarketFac = 0;
	Else If Year>=MarketFacImplemented then MarketFac = 1;
	Else MarketFac=0;

	/*EconNexus DID*/
	If EconNexusImplemented="-" then EconNexus = 0;
	Else If Year>=EconNexusImplemented then EconNexus = 1;
	Else EconNexus=0;

	Keep State Year Social MarketFac EconNexus;
Run;

/*Model 1 - Social w/o Control Variables*/
ods output ParameterEstimates=PEforModel1 DataSummary=ObsModel1
		FitStatistics=AdjRsqModel1 Effects=OverallSigModel1;
```

```
proc surveyreg data=Combine plots=none;
  class State Year / ref=first;
  cluster State;
  model Social = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Model 2 - LogSocial w/o Control Variables*/
data LogCombine;
        set Combine;
        LogSocial = Log(Social);
Run;

ods output ParameterEstimates=PEforModel2 DataSummary=ObsModel2
                FitStatistics=AdjRsqModel2 Effects=OverallSigModel2;
proc surveyreg data=LogCombine plots=none;
  class State Year / ref=first;
  cluster State;
  model LogSocial = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Control Variables Uploading*/

/*Third Data - Party Control*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.PartyControl
```

```
        dbms=xlsx
        replace;
sheet="PartyControl";
getnames=yes;
run;

Proc Sort Data=PartyControl;
        by State;
run;

Proc Transpose Data=PartyControl Out=PartyControl2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PartyControl3;
        Set PartyControl2;
        Year=input(_Name_,4.);
        PartyControl=Col1;
        keep State Year PartyControl;
Run;

/*Fourth Data - GDP*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.GDP
```

```
        dbms=xlsx
        replace;
sheet="GDP";
getnames=yes;
run;

Proc Sort Data=GDP;
        by State;
run;

Proc Transpose Data=GDP Out=GDP2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data GDP3;
        Set GDP2;
        Year=input(_Name_,4.);
        GDP=Col1;
        keep State Year GDP;
Run;

/*Fifth Data - Personal Income*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.PersonalIncome
```

```
        dbms=xlsx
        replace;
sheet="PersonalIncome";
getnames=yes;
run;

Proc Sort Data=PersonalIncome;
        by State;
run;

Proc Transpose Data=PersonalIncome Out=PersonalIncome2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PersonalIncome3;
        Set PersonalIncome2;
        Year=input(_Name_,4.);
        PersonalIncome=Col1;
        keep State Year PersonalIncome;
Run;

/*Sixth Data - Population*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Pop
```

```
        dbms=xlsx
        replace;
sheet="Pop";
getnames=yes;
run;

Proc Sort Data=Pop;
        by State;
run;

Proc Transpose Data=Pop Out=Pop2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data Pop3;
        Set Pop2;
        Year=input(_Name_,4.);
        Pop=Col1;
        keep State Year Pop;
Run;

/*Seventh Data - Social/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.SocialPop
```

```
        dbms=xlsx
        replace;
sheet="SocialPop";
getnames=yes;
run;

Proc Sort Data=SocialPop;
        by State;
run;

Proc Transpose Data=SocialPop Out=SocialPop2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data SocialPop3;
        Set SocialPop2;
        Year=input(_Name_,4.);
        SocialPop=Col1;
        keep State Year SocialPop;
Run;

/*Eighth Data - GDP/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.GDPPop
```

```
        dbms=xlsx
        replace;
sheet="GDPPop";
getnames=yes;
run;

Proc Sort Data=GDPPop;
        by State;
run;

Proc Transpose Data=GDPPop Out=GDPPop2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data GDPPop3;
        Set GDPPop2;
        Year=input(_Name_,4.);
        GDPPop=Col1;
        keep State Year GDPPop;
Run;

/*Ninth Data - Personal Income/Pop*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.PersonalIncomePop
```

```
        dbms=xlsx
        replace;
sheet="PersIncPop";
getnames=yes;
run;

Proc Sort Data=PersonalIncomePop;
        by State;
run;

Proc Transpose Data=PersonalIncomePop Out=PersonalIncomePop2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n;
Run;

Data PersonalIncomePop3;
        Set PersonalIncomePop2;
        Year=input(_Name_,4.);
        PersonalIncomePop=Col1;
        keep State Year PersonalIncomePop;
Run;

/*Tenth Data - Tax Rates*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Rate
```

```
        dbms=xlsx
        replace;
sheet="Rate";
getnames=yes;
run;

Proc Sort Data=Rate;
        by State;
run;

Proc Transpose Data=Rate Out=Rate2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;

Data Rate3;
        Set Rate2;
        Year=input(_Name_,4.);
        Rate=Col1;
        keep State Year Rate;
Run;

/*Eleventh Data - EN Threshold*/
proc import datafile="/home/u63724901/MySAS/HRP/HRPFinalUpload.xlsx"
        out=work.Threshold
```

```
        dbms=xlsx
        replace;
sheet="Threshold";
getnames=yes;
run;

Proc Sort Data=Threshold;
        by State;
run;

Proc Transpose Data=Threshold Out=Threshold2;
        by State;
        var "2008"n "2009"n "2010"n "2011"n "2012"n "2013"n "2014"n "2015"n "2016"n
"2017"n "2018"n "2019"n
"2020"n "2021"n "2022"n "2023"n;
Run;

Data Threshold3;
        Set Threshold2;
        Year=input(_Name_,4.);
        Threshold=Col1;
        keep State Year Threshold;
Run;

/*NEW MODELS*/

/*Combining Everything*/
```

```
/* Step 1: Merge Social3 with Implemented (already done) */
Data Combine;
	Merge Social3 Implemented;
	By State;

	If MarketFacImplemented="-"  then MarketFac = 0;
	Else If Year>=MarketFacImplemented then MarketFac = 1;
	Else MarketFac=0;

	If EconNexusImplemented="-" then EconNexus = 0;
	Else If Year>=EconNexusImplemented then EconNexus = 1;
	Else EconNexus=0;

	keep State Year Social MarketFac EconNexus;
Run;

/* Step 2: Merge Combine with control variables (must sort by State Year first) */
proc sort data=Combine; by State Year; run;
proc sort data=PartyControl3; by State Year; run;
proc sort data=GDP3; by State Year; run;
proc sort data=PersonalIncome3; by State Year; run;
proc sort data=Pop3; by State Year; run;
proc sort data=Rate3; by State Year; run;
proc sort data=Threshold3; by State Year; run;

/* Step 3: Sequential merge */
```

```
data CombineFull;
        merge Combine
            PartyControl3
            GDP3
            PersonalIncome3
            Pop3
            Rate3
            Threshold3;
        by State Year;
run;

/*Model 3 - Social w/ Control Variables*/
ods output ParameterEstimates=PEforModel3 DataSummary=ObsModel3
                FitStatistics=AdjRsqModel3 Effects=OverallSigModel3;
proc surveyreg data=CombineFull plots=none;
  class State Year / ref=first;
  cluster State;
  model Social = MarketFac EconNexus State Year PartyControl GDP PersonalIncome Pop Rate
Threshold / solution adjrsq;
run;
quit;

/*Model 4 - LogSocial w/ Control Variables*/
data LogCombineFull;
        set CombineFull;
        LogSocial = Log(Social);
Run;
```

```
ods output ParameterEstimates=PEforModel4 DataSummary=ObsModel4
                FitStatistics=AdjRsqModel4 Effects=OverallSigModel4;
proc surveyreg data=LogCombineFull plots=none;
  class State Year / ref=first;
  cluster State;
  model LogSocial = MarketFac EconNexus State Year  PartyControl GDP PersonalIncome Pop
Rate Threshold / solution adjrsq;
run;
quit;

/*PER-CAPITA MODELS*/

/*Combine*/
Data CombinePop;
        Merge SocialPop3 Implemented;
        By State;

        /*MarketFac DID*/
        If MarketFacImplemented="-"  then MarketFac = 0;
        Else If Year>=MarketFacImplemented then MarketFac = 1;
        Else MarketFac=0;

        /*EconNexus DID*/
        If EconNexusImplemented="-" then EconNexus = 0;
        Else If Year>=EconNexusImplemented then EconNexus = 1;
        Else EconNexus=0;
```

```
        Keep State Year SocialPop MarketFac EconNexus;
Run;

/*Model 5 - SocialPop w/o Control Variables*/
ods output ParameterEstimates=PEforModel5 DataSummary=ObsModel5
                FitStatistics=AdjRsqModel5 Effects=OverallSigModel5;
proc surveyreg data=CombinePop plots=none;
  class State Year / ref=first;
  cluster State;
  model SocialPop = MarketFac EconNexus State Year / solution adjrsq;
run;
quit;

/*Model 6 - LogSocialPop w/o Control Variables*/
data LogCombinePop;
        set CombinePop;
        LogSocialPop = Log(SocialPop);
Run;

ods output ParameterEstimates=PEforModel6 DataSummary=ObsModel6
                FitStatistics=AdjRsqModel6 Effects=OverallSigModel6;
proc surveyreg data=LogCombinePop plots=none;
  class State Year / ref=first;
  cluster State;
  model LogSocialPop = MarketFac EconNexus State Year / solution adjrsq;
run;
```

```
quit;

/*Control Variables*/
Data CombinePop2;
	Merge SocialPop3 Implemented;
	By State;

	If MarketFacImplemented="-"  then MarketFac = 0;
	Else If Year>=MarketFacImplemented then MarketFac = 1;
	Else MarketFac=0;

	If EconNexusImplemented="-" then EconNexus = 0;
	Else If Year>=EconNexusImplemented then EconNexus = 1;
	Else EconNexus=0;

	keep State Year SocialPop MarketFac EconNexus PartyControl;
Run;

/* Step 2: Merge Combine with control variables (must sort by State Year first) */
proc sort data=CombinePop2; by State Year; run;
proc sort data=PartyControl3; by State Year; run;
proc sort data=GDPPop3; by State Year; run;
proc sort data=PersonalIncomePop3; by State Year; run;
proc sort data=Rate3; by State Year; run;
proc sort data=Threshold3; by State Year; run;

/* Step 3: Sequential merge */
```

```
data CombineFullPop;
        merge CombinePop2
            PartyControl3
            GDPPop3
            PersonalIncomePop3
            Rate3
            Threshold3;
        by State Year;
run;

/*Model 7 - SocialPop w/ Control Variables*/
ods output ParameterEstimates=PEforModel7 DataSummary=ObsModel7
                FitStatistics=AdjRsqModel7 Effects=OverallSigModel7;
proc surveyreg data=CombineFullPop plots=none;
  class State Year / ref=first;
  cluster State;
  model SocialPop = MarketFac EconNexus State Year PartyControl GDPPop
PersonalIncomePop Rate Threshold / solution adjrsq;
run;
quit;

/*Model 8 - LogSocialPop w/ Control Variables*/
data LogCombineFullPop;
        set CombineFullPop;
        LogSocialPop = Log(SocialPop);
Run;
```

```
ods output ParameterEstimates=PEforModel8 DataSummary=ObsModel8
                FitStatistics=AdjRsqModel8 Effects=OverallSigModel8;
proc surveyreg data=LogCombineFullPop plots=none;
  class State Year / ref=first;
  cluster State;
  model LogSocialPop = MarketFac EconNexus State Year PartyControl GDPPop
PersonalIncomePop Rate Threshold / solution adjrsq;
run;
quit;

/*Cleaning*/
Data Table_Long;
        length Model $10;
        length Parameter $30;
        set PEforModel1 PEforModel2 PEforModel3 PEforModel4 PEforModel5 PEforModel6
PEforModel7 PEforModel8 indsname=M;
        ThisIsM=M;

        length Star $3;
        if Probt le 0.01 then Star="***";
                else if Probt le 0.05 then Star="**";
                else if Probt le 0.10 then Star="*";
                else Star="";

        if M="WORK.PEFORMODEL1" then Model="Model1";
                else if M="WORK.PEFORMODEL2" then Model="Model2";
                else if M="WORK.PEFORMODEL3" then Model="Model3";
                else if M="WORK.PEFORMODEL4" then Model="Model4";
```

```
            else if M="WORK.PEFORMODEL5" then Model="Model5";
            else if M="WORK.PEFORMODEL6" then Model="Model6";
            else if M="WORK.PEFORMODEL7" then Model="Model7";
            else if M="WORK.PEFORMODEL8" then Model="Model8";

        Results=Estimate;
        EditedResults=Cats(put(Results,comma16.2),Star);
        output;

        Results=StdErr;
        EditedResults=Cats("(",put(Results,comma16.2),")");
        output;

        keep Model Parameter EditedResults /*Probt Star*/;

run;

proc sort data=Table_Long out=Table_Long_Sorted;
        by Model Parameter;
run;

/* Split cleaned estimates into individual model datasets */
data Model1Results(rename=(EditedResults=Model1));
   set Table_Long_Sorted;
   if Model = "Model1";
   drop Model;
run;
```

```
data Model2Results(rename=(EditedResults=Model2));
    set Table_Long_Sorted;
    if Model = "Model2";
    drop Model;
run;

data Model3Results(rename=(EditedResults=Model3));
    set Table_Long_Sorted;
    if Model = "Model3";
    drop Model;
run;

data Model4Results(rename=(EditedResults=Model4));
    set Table_Long_Sorted;
    if Model = "Model4";
    drop Model;
run;

data Model5Results(rename=(EditedResults=Model5));
    set Table_Long_Sorted;
    if Model = "Model5";
    drop Model;
run;

data Model6Results(rename=(EditedResults=Model6));
    set Table_Long_Sorted;
```

```
    if Model = "Model6";
    drop Model;
run;

data Model7Results(rename=(EditedResults=Model7));
    set Table_Long_Sorted;
    if Model = "Model7";
    drop Model;
run;

data Model8Results(rename=(EditedResults=Model8));
    set Table_Long_Sorted;
    if Model = "Model8";
    drop Model;
run;

data Table_Wide;
        merge Model1Results Model2Results Model3Results Model4Results Model5Results
Model6Results Model7Results Model8Results;
        by Parameter;

        length Order 3;
        if Parameter="week" then Order=1;
                else if Parameter="age" then Order=2;
                else Order=3;

        if mod(_n_,2)=1 then Regressors=Parameter;
```

```
run;

proc sort data=Table_Wide out=Table_Wide_Sorted(drop=Order Parameter);
	by Order;
run;

Data Table_Long;
	length Model $10;
	length Parameter $30;
	set PEforModel1 PEforModel2 PEforModel3 PEforModel4 PEforModel5 PEforModel6
PEforModel7 PEforModel8 indsname=M;
	ThisIsM=M;
	Where (Estimate ne 0) and (substr(Parameter,1,5) ne "State") and (substr(Parameter,1,4)
ne "Year");

	length Star $3;
	if Probt le 0.01 then Star="***";
		else if Probt le 0.05 then Star="**";
		else if Probt le 0.10 then Star="*";
		else Star="";

	if M="WORK.PEFORMODEL1" then Model="Model1";
		else if M="WORK.PEFORMODEL2" then Model="Model2";
		else if M="WORK.PEFORMODEL3" then Model="Model3";
		else if M="WORK.PEFORMODEL4" then Model="Model4";
		else if M="WORK.PEFORMODEL5" then Model="Model5";
		else if M="WORK.PEFORMODEL6" then Model="Model6";
```

```
            else if M="WORK.PEFORMODEL7" then Model="Model7";
            else if M="WORK.PEFORMODEL8" then Model="Model8";

        Results=Estimate;
        EditedResults=Cats(put(Results,comma16.4),Star);
        output;

        Results=StdErr;
        EditedResults=Cats("(",put(Results,comma16.4),")");
        output;

        keep Model Parameter EditedResults /*Probt Star*/;

run;

proc sort data=Table_Long out=Table_Long_Sorted;
        by Model Parameter;
run;

data _null_;
   put "STARTING CLEAN SPLIT OF MODEL RESULTS";
run;

data
   Model1Results(rename=(EditedResults=Model1))
   Model2Results(rename=(EditedResults=Model2))
   Model3Results(rename=(EditedResults=Model3))
```

```
    Model4Results(rename=(EditedResults=Model4))
    Model5Results(rename=(EditedResults=Model5))
    Model6Results(rename=(EditedResults=Model6))
    Model7Results(rename=(EditedResults=Model7))
    Model8Results(rename=(EditedResults=Model8));

    set Table_Long_Sorted;

    select (Model);
        when ("Model1") output Model1Results;
        when ("Model2") output Model2Results;
        when ("Model3") output Model3Results;
        when ("Model4") output Model4Results;
        when ("Model5") output Model5Results;
        when ("Model6") output Model6Results;
        when ("Model7") output Model7Results;
        when ("Model8") output Model8Results;
        otherwise;
    end;

    drop Model;
run;

/*Ordering*/
data Table_Wide;
        merge Model1Results Model2Results Model3Results Model4Results Model5Results
Model6Results Model7Results Model8Results;
```

```
	by Parameter;

	length Order 3;
	if Parameter="DID" then order=1;
		else if Parameter="State" then order=2;
		else if Parameter="Year" then order=3;
		else if Parameter="Intercept" then order=4;

	if mod(_n_,2)=1 then Regressors=Parameter;

run;

proc sort data=Table_Wide out=Table_Wide_Sorted(drop=Order Parameter);
	by Order;
run;

/*Rows for other statistics and info*/
/*Number of Obs.*/

/* Filter and format number of observations for each model */
data Obs1; set ObsModel1; if Label1="Number of Observations"; Model1 = put(NValue1,
comma16.); keep Label1 Model1; run;
data Obs2; set ObsModel2; if Label1="Number of Observations"; Model2 = put(NValue1,
comma16.); keep Label1 Model2; run;
data Obs3; set ObsModel3; if Label1="Number of Observations"; Model3 = put(NValue1,
comma16.); keep Label1 Model3; run;
data Obs4; set ObsModel4; if Label1="Number of Observations"; Model4 = put(NValue1,
comma16.); keep Label1 Model4; run;
```

```
data Obs5; set ObsModel5; if Label1="Number of Observations"; Model5 = put(NValue1,
comma16.); keep Label1 Model5; run;

data Obs6; set ObsModel6; if Label1="Number of Observations"; Model6 = put(NValue1,
comma16.); keep Label1 Model6; run;

data Obs7; set ObsModel7; if Label1="Number of Observations"; Model7 = put(NValue1,
comma16.); keep Label1 Model7; run;

data Obs8; set ObsModel8; if Label1="Number of Observations"; Model8 = put(NValue1,
comma16.); keep Label1 Model8; run;


/* Merge all */

data NumofObs;

        merge Obs1 Obs2 Obs3 Obs4 Obs5 Obs6 Obs7 Obs8;

        by Label1;

run;


/*Adjusted R-Square*/


/* Filter and rename each Adjusted R-Square dataset individually */

data AdjRsq1; set AdjRsqModel1 (rename=(CValue1=Model1) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq2; set AdjRsqModel2 (rename=(CValue1=Model2) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq3; set AdjRsqModel3 (rename=(CValue1=Model3) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq4; set AdjRsqModel4 (rename=(CValue1=Model4) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq5; set AdjRsqModel5 (rename=(CValue1=Model5) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq6; set AdjRsqModel6 (rename=(CValue1=Model6) drop=NValue1); if
Label1="Adjusted R-Square"; run;
```

```
data AdjRsq7; set AdjRsqModel7 (rename=(CValue1=Model7) drop=NValue1); if
Label1="Adjusted R-Square"; run;

data AdjRsq8; set AdjRsqModel8 (rename=(CValue1=Model8) drop=NValue1); if
Label1="Adjusted R-Square"; run;


/* Merge all together */

data AdjRsq;

        merge AdjRsq1 AdjRsq2 AdjRsq3 AdjRsq4 AdjRsq5 AdjRsq6 AdjRsq7 AdjRsq8;

        by Label1;

run;


/*Overall Significance (F-Test)*/


/* Step 1: Filter and format each model's F-test result */

data OSM1; set OverallSigModel1; if Effect="Model"; Label1="Overall Significance";

   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";

   Model1 = cats(put(Fvalue, comma20.2), Star);

   keep Label1 Model1;

run;


data OSM2; set OverallSigModel2; if Effect="Model"; Label1="Overall Significance";

   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if
Probf<=0.10 then Star="*"; else Star="";

   Model2 = cats(put(Fvalue, comma20.2), Star);

   keep Label1 Model2;

run;
```

```
data OSM3; set OverallSigModel3; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model3 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model3;
run;

data OSM4; set OverallSigModel4; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model4 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model4;
run;

data OSM5; set OverallSigModel5; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model5 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model5;
run;

data OSM6; set OverallSigModel6; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model6 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model6;
run;
```

```
data OSM7; set OverallSigModel7; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model7 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model7;
run;

data OSM8; set OverallSigModel8; if Effect="Model"; Label1="Overall Significance";
   length Star $3; if Probf<=0.01 then Star="***"; else if Probf<=0.05 then Star="**"; else if Probf<=0.10 then Star="*"; else Star="";
   Model8 = cats(put(Fvalue, comma20.2), Star);
   keep Label1 Model8;
run;

/* Step 2: Merge all formatted outputs */
data OverallSig;
   merge OSM1 OSM2 OSM3 OSM4 OSM5 OSM6 OSM7 OSM8;
   by Label1;
run;

/*Combine F-values*/
Data OverallSig;
        merge OSM1 OSM2 OSM3 OSM4 OSM5 OSM6 OSM7 OSM8;
run;

/*Controls*/
Data Controls;
        length Label1 $30;
```

```
        Label1="Controls for State?";
        Model1="Yes"; Model2="Yes"; Model3="Yes"; Model4="Yes";
        Model5="Yes"; Model6="Yes"; Model7="Yes"; Model8="Yes";
        output;

        Label1="Controls for Year?";
        output;
run;

/*Combine Rows*/
Data OtherStat;
        set NumofObs AdjRsq OverallSig Controls;
        rename Label1=Regressors;
run;

/*Add Other Statistics to Results Table*/

Data Table_wide_sorted_withstat;
        set Table_Wide_Sorted OtherStat;
run;

/* Print Cleaned Table*/
ods excel file="/home/u63724901/MySAS/AllModelsCleanedHRP_Social.xlsx"
options(Embedded_Titles="ON" Embedded_Footnotes="ON");
Title "Title";
footnote1 justify=left "Sources: Source";
```

```
footnote2 justify=left "Notes: robust standard errors are in parentheses. *, **, and ***
indicate 10%, 5%, and 1% significance levels, respectively.";
proc print data=Table_Wide_Sorted_withstat noobs;
        var regressors;
        var Model1-Model8 / Style(header)={just=center} style(data)={just=center};
        format Regressors $VariableName.;
run;
ods excel close;

/*//////////////Parallel Trends/////////////////*/
/* STEP 1: Merge SocialPop with Implemented */
data CombineSocialPop;
   merge SocialPop3 Implemented;
   by State;

   /* MarketFac & EconNexus DIDs */
   if MarketFacImplemented="-" then MarketFac = 0;
   else if Year >= MarketFacImplemented then MarketFac = 1;
   else MarketFac = 0;

   if EconNexusImplemented="-" then EconNexus = 0;
   else if Year >= EconNexusImplemented then EconNexus = 1;
   else EconNexus = 0;
run;

/* STEP 2: Assign treatment and relative time */
data CombineSocialPop2;
```

```
    set CombineSocialPop;

    /* Control group: states without sales tax */
    if State in (2, 8, 26, 29, 37) then treatment = 0;
    else treatment = 1;

    /* Relative time from 2018 */
    rt = Year - 2018;
run;

/* STEP 3: Run parallel trends test (pre-treatment only) */
ods output ParameterEstimates=PE_ParallelTrends_SocialPop
           FitStatistics=Fit_ParallelTrends_SocialPop;

proc surveyreg data=CombineSocialPop2;
    where Year < 2018;
    class State;
    model SocialPop = rt rt*rt rt*rt*rt
                      treatment treatment*rt treatment*rt*rt treatment*rt*rt*rt
                      State
                      / solution adjrsq;
run;

ods output close;

/* STEP 4: Export to Excel */
ods excel file="/home/u63724901/MySAS/ParallelTrends_SocialPop.xlsx";
```

```
ods excel options(sheet_name="Parameter Estimates");
proc print data=PE_ParallelTrends_SocialPop noobs label;
   title "Parallel Trends Test – Per Capita Social";
run;

ods excel options(sheet_name="Fit Statistics");
proc print data=Fit_ParallelTrends_SocialPop noobs label;
   title "Fit Statistics – Per Capita Social";
run;

ods excel close;

/*Summary Stats*/
ods excel file="/home/u63724901/MySAS/SocialPop_Summary_All.xlsx"
options(sheet_interval='none');

/* Sheet 1: Overall Summary */
ods excel options(sheet_name='Overall Summary');
proc means data=SocialPop3 n mean median std min max;
   var SocialPop;
run;

/* Sheet 2: By State */
ods excel options(sheet_name='By State');
proc means data=SocialPop3 n mean median std min max;
   class State;
```

```
    var SocialPop;
run;

/* Sheet 3: By Year */
ods excel options(sheet_name='By Year');
proc means data=SocialPop3 n mean median std min max;
    class Year;
    var SocialPop;
run;

ods excel close;
```