Business School

THE IMPACT OF CHANGES IN PARTY POLITICAL CONTROL IN ENGLAND ON CARE HOME QUAILITY

Sigao Li
2638795

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in Business Analytics in the Faculty of Social Science and Law.

University of Bristol Business School
August 2025

Word count: 9863

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Taught Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, this work is my own work. Work done in collaboration with, or with the assistance of others, is indicated as such. I have identified all material in this dissertation which is not my own work through appropriate referencing and acknowledgement. Where I have quoted or otherwise incorporated material which is the work of others, I have included the source in the references. Any views expressed in the dissertation, other than referenced material, are those of the author.

I have read and understand the University's Academic Integrity Policy including plagiarism, contract cheating, falsifying data and breaching ethical research protocols.

SIGNED: Sigao Li DATE: 2025.08.29

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Wen Zhang, for her invaluable support, expertise, guidance, and patience throughout the process of writing this major research paper. Without her insightful guidance and continuous encouragement, this research would have been far more challenging. Additionally, I would like to extend my appreciation to Dr. Sunil Tiwari for serving as my approved academic checker; his thoughtful questions and constructive suggestions were greatly valued.

I am also thankful to Dr. Minhao Zhang from the University of Bristol and Dr. Mike Tse from Cardiff University. Although not directly supervising my Dissertation, their valuable insights and expert advice significantly enriched my understanding of this research topic.

Finally, I would like to express my appreciation to all faculty members who supervise and teach in the Business Analytics program, whose guidance and support have been instrumental throughout my graduate studies at the University of Bristol.

Abstract

This study investigates the influence of local political dynamics on the quality of care homes in England between 2016 and 2023. Drawing on a balanced panel of 116 local authorities, it integrates Care Quality Commission ratings with local government financial and socioeconomic data. Employing a parallel process latent growth model within a Bayesian framework, the analysis tests both the direct effects of long-term party control and party alternation, as well as the mediating role of per capita adult social care expenditure. The findings reveal that long-term partisan structures exert no statistically significant influence on care home quality, suggesting a potential tendency toward political inertia under stable governance. By contrast, political alternation, specifically a transition from Conservative to Labour control, has a robust and positive effect on quality improvement. Importantly, these effects are not mediated by changes in expenditure, indicating that fiscal input alone does not explain political impacts on service outcomes. Instead, the administrative shock of turnover emerges as the key mechanism driving change. The study contributes to debates on public service performance by challenging the assumption of a "spending–quality" pathway and highlighting the independent role of political reorientation. Beyond the UK, the findings have comparative relevance for welfare states with decentralized governance, offering insights into how local politics shape institutional quality and social care provision.

Keywords: Care home quality; Adult social care; Local politics; Bayesian analysis

Table of Contents

List of Tables

List of Figures

List of Appendices

Chapter 1: Introduction

1.1 Statement of the Research Problem

The adult social care system in the United Kingdom is facing a profound systemic crisis rooted in long-standing funding problems and policy neglect (Bliss, 2022; Merritt, 2024). Since 2010, austerity policies have led to substantial cuts in central government grants for local authorities, which are the primary funders and providers of social care services (Gray and Barford, 2018; Local Government Association, 2025). Although nominal spending has increased, this growth has mainly been driven by demographic pressures and rising provider costs, rather than by efforts to improve service quality or expand coverage (Ogden and Phillips, 2024; Foster and Harker, 2025). Continued fiscal restraint has forced local authorities to make difficult choices in fulfilling their statutory duties, with preventive but non-statutory services often the first to be reduced(Cummins, 2018; Gray and Barford, 2018; Alonso and Andrews, 2020; Iparraguirre, 2020; Stokes et al., 2022).

Against this backdrop, the political environment of local authorities has become a key variable in resource allocation, yet its influence on care home quality is more complex than commonly assumed. The existing literature on care home quality has primarily focused on external market conditions and internal management mechanisms (Chang and Cheng, 2013; Forder and Allan, 2014; Dellefield et al., 2015; Yang et al., 2022). While political science research has shown that governing party ideology shapes public services(Hall, 1993; Wang et al., 2020), and has often identified fiscal spending as a central factor affecting service quality (Bradley et al., 2011; Cardona et al., 2021; Oronce et al., 2025). Yet, few studies have directly tested the "party–spending–quality" pathway empirically.

This study aims to fill the research gap by developing a mediation analysis framework to systematically investigate the impact of local politics on the quality of care homes. Specifically, it quantifies the direct impact of long-term party control and party alternation in English local authorities on care home quality, while testing whether per capita adult social care expenditure serves as a mediating factor. The analysis utilizes a balanced panel dataset comprising 116 local authorities from 2016 to 2023, which combines official ratings from the Care Quality Commission (CQC) with local government financial data. It also controls for macroeconomic conditions, demographic structure, and the impact of the COVID-19 pandemic. Methodologically, following Webb and Bywaters (2024), local political dynamics during the study period are modelled. Adult social care expenditure per capita is then introduced as a mediator. This allows testing whether financial input is the key link between politics and service quality, and estimating the total effect of political factors. Within this framework, two dimensions of political influence are central: (i) the long-term partisan structure of local authorities, reflecting governance continuity; and (ii) shifts in party control, understood as moments of political reorientation. Preventive spending is conceptualized as the mediating channel through which politics may affect quality. This design leads to four testable propositions: that long-term partisan structures directly shape care home quality (H1), and may do so indirectly via preventive spending (H2); and that party alternation affects quality directly (H3), while also potentially working through preventive spending (H4).

To preface the results, this study presents a complex picture. First, party alternation has a significant direct effect on care home quality. However, this effect is not transmitted through changes in adult social care expenditure. In other words, the "administrative shock" of political turnover itself, rather than budgetary adjustments, is the key driver of quality improvements. Second, the effect of long-term party control is not statistically significant. Yet, its direction may suggest a tendency toward "political inertia" (Michels et al., 1915), in which stable political structures reduce incentives for change and reinforce policy continuity. These results provide important insights into how local politics shape social care services and raise new questions for future research on non-fiscal pathways of political influence. Although the analysis is situated in the United Kingdom, the mechanisms under investigation (party control, fiscal priorities, and service quality) are also present in other welfare states with decentralized governance (Rodden et al., 2003; Borge et al., 2008; Grembi et al., 2016). Therefore, the findings have comparative and international relevance, offering lessons that extend beyond the UK context.

In sum, this study makes a substantial contribution to the fields of public health and policy administration. It is the first to distinguish and test the different effects of long-term party control and political alternation on care home quality, while challenging the widely assumed "fiscal expenditure pathway." Unlike previous studies that either broadly examined the link between spending and service quality (Grabowski, 2004; Boyne et al., 2012), or focused on other public services such as child welfare (Alonso and Andrews, 2020; Webb and Bywaters, 2024). This study explicitly centers on care homes, a sector heavily reliant on public regulation and funding, and investigates the mediating role of "preventive expenditure." In doing so, it addresses a key limitation in the existing literature concerning the mechanisms through which politics influences service outcomes.

1.2 Organization of the Paper

This study is organized into seven chapters, each of which builds upon the preceding one to provide a coherent and comprehensive exploration of the research topic. Following the introduction, Chapter Two provides a comprehensive review of the literature on care home quality assessment and the influence of political parties on public services, which is essential for understanding the mechanisms behind party control or alternation and their impact on care home quality. Chapter Three then provides a detailed account of the data sources, variables, and methodological strategies employed in the analysis, ensuring transparency and rigour in the research design. Chapter Four presents the empirical findings, highlighting key patterns and relationships, while also conducting a series of robustness checks and sensitivity analyses to demonstrate the reliability and validity of the results. Chapter Five extends beyond the empirical results to discuss the theoretical implications of the study, considering how the findings contribute to debates in political science and public health, as well as their relevance for policymakers seeking to improve care home standards. Chapter Six concludes the study by summarizing the main insights, acknowledging the study's limitations, and offering recommendations for both future research and practical policy reform.

Chapter 2: Literature Review

This chapter aims to develop the conceptual framework for the study by reviewing relevant research. It is divided into three sections. The first section systematically reviews the existing literature on the assessment of care home quality, with a particular focus on research conducted in England. The second section examines the relationship between political factors and public service performance, focusing on how party alternation can create policy discontinuity, how local party ideology shapes the provision of public services, and how these political differences generate regional variation in social care quality. The third section builds on the literature review to identify research gaps, propose a theoretical framework, and outline the hypotheses that guide the subsequent empirical analysis.

2.1 Assessing and Understanding the Quality of Care Homes

Against the backdrop of global population aging, the quality of adult social care services, particularly in care homes, has become a central topic of scholarly inquiry (Netten et al., 2012; Forder and Allan, 2014; Hudson, 2019). Existing research generally classifies the influencing factors into two broad dimensions: the external market environment and internal management mechanisms. To avoid conceptual ambiguity and align with health services research, this article interprets "quality" as a multidimensional construct, following the structure-process-outcome framework (Donabedian, 1988). This will allow for a clearer identification of indicators and causal pathways in subsequent empirical research.

From the perspective of the external environment, market structure, ownership type, and institutional scale are regarded as three critical determinants of nursing home quality. Some studies measure market concentration and price levels to examine their impact on care quality (Knapp et al., 2001; Forder and Allan, 2014; Yang et al., 2022). Dintrans (2018) highlights that nursing homes often exhibit spatial clustering, especially in urban centers with higher levels of aging and household income. Such concentration inevitably affects market share. To compete for residents, providers may adopt price-based strategies, which can reduce revenue per unit and strain resources, ultimately undermining service quality (Forder and Allan, 2014). Conversely, nursing homes operating in less competitive markets face lower risks of closure (Allan and Forder, 2015). Thus, the competitive dynamics of local markets set important structural constraints that shape providers' strategic choices and quality outcomes.

Ownership models (for-profit or non-profit) represent another key factor. A substantial body of empirical evidence demonstrates systematic differences in both process and outcome quality between the two. For-profit institutions are often associated with lower levels of care (Harrington et al., 2001; Hillmer et al., 2005; Comondore et al., 2009). However, compared with ownership, the impact of institutional scale is more nuanced and presents a more complex picture. Baldwin et al. (2017) report that smaller facilities are more likely to deliver higher-quality care compared with larger institutions with more bed capacity. On the other hand, smaller care homes also face a higher risk of closure due to staff shortages and are increasingly perceived as less viable (Netten, Darton and Williams, 2003). Taken together, these three dimensions (market competition, ownership, and scale) highlight that external structural conditions can both enable and constrain the pursuit of high-quality care.

Corresponding to the external environment, the internal management mechanisms of care homes, particularly performance evaluation, human resource management, and operational efficiency, are regarded as direct determinants of care quality. With respect to performance evaluation, research suggests that information transparency is a crucial instrument for improving quality. In the absence of effective regulation and quality control, service quality often deteriorates (Harrington et al., 2014). This is especially evident in the nursing home sector. Data show that closures frequently result from violations of safety standards, leading to compulsory shutdowns (Iqbal et al., 2023; Bach-Mortensen et al., 2024). Public reporting of performance strengthens external accountability, compelling providers to improve service quality (Stevenson, 2006), while also fostering internal motivation for quality enhancement (Fung et al., 2008). Such transparency provides consumers with valuable information when selecting providers. Consumer-driven market feedback, in turn, creates competitive pressure that pushes providers to raise their standards (Werner et al., 2012). This suggests that internal monitoring and external accountability are closely intertwined, jointly reinforcing incentives for quality improvement. In the English context, CQC inspections and ratings provide a key vehicle for this "information transparency-market feedback" mechanism. As the independent regulator of health and adult social care services, CQC plays a crucial role in assessing the quality of care homes. Its standardized and authoritative rating system is widely adopted in academic research as a measure of service quality (Barron and West, 2017; Bach-Mortensen et al., 2024).

In contrast, workforce-related factors are often considered the most direct and central drivers of care quality. Literature consistently underscores the importance of human resource management. For example, higher staffing levels enable more personalized and timely care (Dellefield et al., 2015); lower staff turnover enhances continuity and professionalism of services (Castle and Anderson, 2011); and ongoing training and skill development are essential for addressing complex care needs (Antwi and Bowblis, 2018). These factors directly shape the quality of care homes. Yet, even with sufficient staffing, how efficiently resources are deployed remains a critical issue. The relationship between operational efficiency and quality is more complex and bidirectional. Evidence suggests that quality has a significant impact on efficiency (Chang and Cheng, 2013), while inefficiency increases operating costs (Dormont and Martin, 2012). However, the association between different quality dimensions and technical efficiency is not uniform and may even involve trade-offs (Laine et al., 2005; Dulal, 2018).

Taken together, these studies provide valuable insights by examining nursing home quality from both external and internal perspectives, demonstrating that multiple, interrelated factors shape service quality. Nonetheless, existing research has largely overlooked the influence of the political environment on nursing home quality.

2.2 The Relationship between Political Parties and Public Service

Current research consistently shows a close connection between political parties and the quality of public services. Parties exert influence primarily through shaping policy priorities and fiscal budgets, thereby indirectly affecting service outcomes (Dawkins, 2017; Wang et al., 2020; Di et al., 2022). Different parties often carry distinct ideological positions and policy commitments. When in power, they may introduce new policy agendas or deliberately dismantle the core policies of their predecessors, thus reshaping policy trajectories and producing divergent outcomes (Hall, 1993; Häusermann et al., 2013). To appeal to voters, some parties favour projects with immediate results, while showing little interest in costly and slow-impact "preventive policies," such as public health, basic education, and social care reform (Jacobs, 2011; Cairney and St Denny, 2020; Ogami, 2024). In the United Kingdom, the austerity program implemented under Conservative leadership since 2010 has significantly deepened regional inequalities and weakened the capacity of some local governments to provide public services (Gray and Barford, 2018). Evidence of declining service quality has been documented in several areas, including child welfare (Alonso and Andrews, 2020), mental health provision (Cummins, 2018), and elderly care (Iparraguirre, 2020).

However, this trend is not uniform. Like many other countries, the United Kingdom has sought to promote “local empowerment” by devolving greater policy responsibility to local governments, thereby fostering more stable reforms that are responsive to citizens’ needs (Hendriks et al., 2010; Yi and Qiu, 2025). In practice, however, local execution in public services is often constrained by strategic ambiguity, limited financial resources, and uneven governance capacity (Boyne and Dahya, 2002; Liddle et al., 2022). In other words, while decentralization is expected to enhance responsiveness, its effectiveness depends heavily on the fiscal and administrative environment within which local governments operate. Despite these challenges, both major parties (Conservatives and Labour) have recently expressed support for empowering local authorities (The Labour Party, 2022), expecting such measures to stimulate local policy innovation (HM Government, 2022).

As greater discretion over social protection shifts to the local level, scholars have increasingly asked whether the political orientation of local parties influences the quality of public services (Alonso and Andrews, 2020; Hick, 2022; Webb and Bywaters, 2024). The most evident channel lies in differences in budget allocation and fiscal priorities, which shape resources devoted to social services (Bradley et al., 2016). For example, left-leaning parties are generally more willing to expand welfare coverage through higher investment in social provision, whereas right-leaning parties emphasize market mechanisms and liberalization in welfare governance (Fernández-I-Marín et al., 2024). Hick (2022) points out that, even after controlling for demographic and economic factors, councils controlled by the Conservative Party are markedly more likely to cut social service budgets. Comparable political effects are observed internationally. Residents of conservative U.S. states experience notably poorer overall health outcomes than those in liberal states (Krieger et al., 2024). These studies indirectly prove that there is a positive association between higher spending on social care and service quality (Grabowski, 2004; Bradley et al., 2011; Cardona et al., 2021; Oronce et al., 2025). However, it's important to note that this partisan effect doesn't materialize overnight and often exhibits significant path dependence (Pierson, 2000). Once existing budget structures, regulatory paradigms, and organizational capacity are internalized, they have a cumulative and self-reinforcing effect on subsequent policy redistribution and quality governance. Therefore, the influence of partisans depends both on current choices and on historical trajectories.

These studies suggest that party influence does not operate through a single mechanism. Instead, partisan orientation may indirectly shape service quality through its effects on policy priorities, budgetary redistribution, and even governance structures. This highlights the need for theoretical and methodological approaches that move beyond simple “party dummy variables” toward identifying the underlying mechanisms of influence.

2.3 Research Framework and Hypotheses

Although existing research has examined the role of external market environments (Forder and Allan, 2014; Knapp et al., 2001), and internal management mechanisms (Stevenson, 2006; Castle and Anderson, 2011; Dellefield et al., 2015) in shaping care home quality, the political environment in which care homes operate has not been incorporated into analytical frameworks. This omission is important because the literature on political parties and public services shows that parties influence not only policy priorities and budget allocation (Hall, 1993; Häusermann et al., 2013; Wang et al., 2020), but also the quality of public services through spending structures (Bradley et al., 2011; Hick, 2022; Fernández-I-Marín et al., 2024). In the UK, the simultaneous pressures of fiscal austerity and local empowerment highlight the potential link between local party politics and the quality of social care (Gray and Barford, 2018; Liddle et al., 2022). Yet despite these indications, care home quality, arguably the most direct determinant of older people’s wellbeing within adult social care, remains largely absent from research that connects it to party politics.

At least three gaps can be identified in the existing literature. First, despite clear evidence of party influence on public services, research has focused primarily on areas such as child welfare, mental health, or aggregate social care spending (Boyne et al., 2012; Alonso and Andrews, 2020; Iparraguirre, 2020), while neglecting care homes, a sector highly dependent on public regulation and financial support. Second, although the fiscal effects of party politics are well established (Grabowski, 2004; Cardona et al., 2021), the mechanisms of expenditure have not been sufficiently disaggregated. In particular, the role of "preventive spending" in creating supply-side pressures that shape care home performance has not been systematically examined. Third, most studies emphasize the static effect of long-term party control, while paying less attention to the dynamic impact of party change. Yet shifts in party control often entail a reconfiguration of policy momentum (Pierson, 2000), which may independently affect care home quality by altering contracts, regulatory priorities, or the timing of fiscal allocations. These three gaps together suggest that a more integrated framework is needed to connect party politics with care home outcomes.

Therefore, although no study has directly examined the relationship between political parties and care home quality, prior evidence suggests that parties influence other areas of public service outcomes (Webb and Bywaters, 2024), providing a way forward. By applying a rigorous mediation framework, this study seeks to fill that gap. Building on this insight, this study develops a "party–finance–quality" theoretical model (see Figure 1) to systematically investigate the causal mechanisms linking local politics to the quality of care homes. Within this framework, care home quality is conceptualized as the ultimate performance outcome, shaped by two core political factors: the long-term partisan structure of local authorities, which reflects continuity in governance orientation, and shifts in party control, understood as signals of policy reorientation. To trace the transmission channels of political influence, preventive spending is introduced as a central mediating variable. This design enables testing the extent to which party politics indirectly affect care home quality by shaping fiscal priorities. In addition, to ensure robust estimates, the framework controls for socioeconomic conditions such as population density, income per capita, and unemployment.

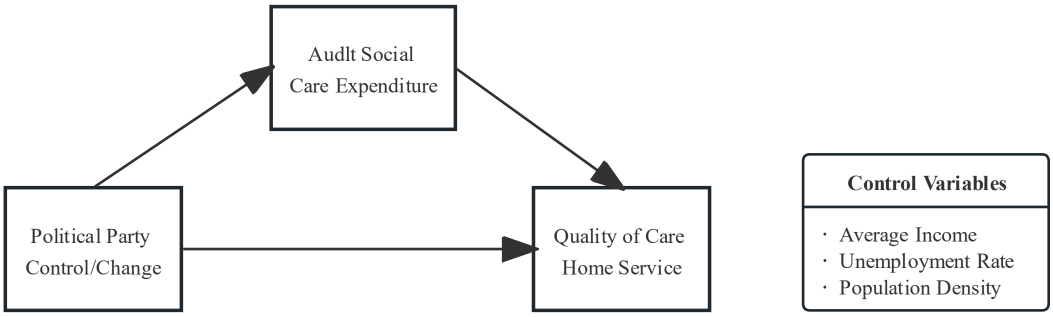

Figure 1. Mediation Framework Linking Political Affiliation and Care Home Quality

Based on this framework, the study proposes four hypotheses:

H1. The long-term partisan structure of local authorities affects care home quality.

H2. The long-term partisan structure indirectly affects care home quality through preventive spending.

H3. Shifts in local party control affect care home quality.

H4. Shifts in local party control indirectly affect care home quality through preventive spending.

Taken together, these hypotheses form a coherent analytical system. H1 highlights the role of governance continuity in determining the depth and duration of quality improvements. H2 identifies preventive spending as a key transmission mechanism. H3 incorporates political dynamics into the framework, emphasizing the independent contribution of policy reorientation to quality trajectories. H4 tests whether preventive spending mediates the effect of political change. Collectively, these hypotheses provide the bridge between theory and empirical investigation, linking the questions of “who governs,” “how they govern,” and “whether priorities shift” to “how quality evolves over time,” thereby providing a clear theoretical foundation and testable propositions for the methodological and empirical sections that follow.

Chapter 3: Research Design and Methodology

This chapter presents the research design and methodological framework, with a focus on the relationship between local politics and care home quality, as well as its operationalization. First, it outlines the data sources, variable construction, and measurement strategies, covering party control, CQC quality ratings, fiscal expenditure, and a set of socioeconomic control variables. It also explains the principles of data cleaning and standardization. Second, it introduces the parallel process latent growth model (PPLGM), highlighting its suitability for modelling dynamic processes and testing mediation effects, and discusses the advantages of Bayesian estimation in handling small samples and complex hierarchical structures. Finally, it addresses the standards of inference and robustness checks adopted to ensure credibility and interpretability of the findings. Together, these elements provide the empirical foundation and methodological support for the subsequent analysis.

3.1 Variable Construction and Measurement

The data for this study were drawn from multiple public databases, covering the fiscal years 2016 to 2023 (see Table 1). This period encompasses significant political and economic fluctuations in the United Kingdom (Fieldhouse and Bailey, 2023), and spans the years preceding and following the outbreak of the COVID-19 pandemic. Such a timeframe enables the analysis to carefully account for the pandemic's disruptive impact when examining the influence of local political factors on care homes.

Table 1. Data Sources

Data Name	Source	Year	Type
Local Elections	UK Parliament	2015-2022	TEXT
Care Homes Inspection Reports	Care Quality Commission	2016-2023	TEXT
Local Authority Expenditure	GOV UK	2016-2023	TEXT
Annual Survey of Hours and Earnings	Office for National Statistics	2016-2023	TEXT
Labour Market in The Regions of the UK	Office for National Statistics	2016-2023	TEXT
Consumer Price Inflation	Office for National Statistics	2016-2023	TEXT
Population Estimates	Office for National Statistics	2016-2023	TEXT
Counties and Unitary Authorities Boundaries	Office for National Statistics	2016-2023	SHP

The key explanatory variable captures changes in political control of local authorities. Annual election results for all local authorities from 2015 to 2023 were manually extracted and cross-checked using the official UK Parliament website (UK Parliament, 2025). Following the approach of (Boyne et al., 2012), the political variable was lagged by one year. For example, data on party control in 2016 are based on the 2015 election results. This adjustment reflects the fact that local authority budgets are typically drafted and approved before election day, which limits the capacity of newly elected administrations to make substantial changes within the same fiscal year (Local Government Association, 2024; Sandford, 2025) . The variable was operationalized as a three-category nominal measure of the ruling party, with the Conservative Party serving as the reference group (coded 1), and the Labour Party (0) and No Overall Control (2) as the other two categories.

The dependent variable is the average overall quality rating from CQC. These data were obtained via the official CQC API (Care Quality Commission, 2025), which provides the complete dataset of all registered care homes in England. The dataset includes detailed inspection reports, quality scores, and records of registration and deregistration. Based on this source, average overall CQC scores were calculated for all inspected care homes within each local authority.

Additionally, to account for structural factors that may influence trajectories of change, three time-varying controls were included: unemployment rate, population density, and average annual income. These variables are designed to account for macroeconomic and social differences over time. The first control variable is the unemployment rate, a standard indicator of local economic conditions (O'Toole and Meier, 1999). Deteriorating economic conditions are expected to increase demand for social services, creating additional challenges that make it more difficult to maintain high service quality (Boyne et al., 2012). Including this factor also helps address regional disparities arising from lower care quality in more deprived areas (Park and Martin, 2018). However, because official unemployment data are unavailable at the upper-tier local authority level, the study follows the Bureau of Labor Statistics (2025), applies a proxy measure. Specifically, the unemployment rate was approximated using the Labour Market in the Regions of the UK dataset (Office for National Statistics, 2025a), calculated as (total unemployment benefit claimants ÷ adult population) × 100. Population data were obtained from the Office for National Statistics' nomis platform. These figures were combined with regional area data to compute population density, capturing the degree of spatial concentration and market size. This measure also reflects the clustering effects observed in the geographical distribution of care homes (Dintrans, 2018). Regional average annual income was taken from the Annual Survey of Hours and Earnings (Office for National Statistics, 2024).

The mediating variable is local government expenditure on adult social care, drawn from the Local Authority Expenditure dataset (GOV UK, 2025). Following Webb and Bywaters (2024), comparability across regions was achieved by dividing each authority's annual adult social care expenditure by its adult population.

Finally, to explicitly capture the common environmental shock of the COVID-19 pandemic (Ng, 2021), two dummy variables were introduced following Zeng et al. (2024). The first represents a pulse effect in 2020 (Pulse = 1, other years = 0). The second captures an exponential recovery trend across 2021–2023 (Recovery = 1, 2, 3 for those years, and 0 for all other years). This specification enables the model to identify the abrupt level shift in 2020 and the subsequent recovery trajectory (see Table 2).

Table 2. Variables

Variable Name	Description
Plan_P	The party that sets the spending plan for the year
Overall	The average overall scores of care homes by CQC inspection in the region
Unemp_Rate	The percentage of the population aged 18+ who filed for claimant benefits in the year
Mean	Average annual gross income (£)
P_Density	Population density of the region
LATE_P18	Local authority spending per person aged 18+ on adult social care (£)
Covid	The period during which COVID-19 occurred

To ensure consistency across both spatial and temporal dimensions, all datasets were matched to official ONS codes and corresponding years using boundary files provided by the Office for National Statistics (2025c). The data were then merged into a single panel dataset. To satisfy the identification requirements of the parallel growth model, only cases with complete observations for every year from 2016 to 2023 were retained. During this process, local authorities that had undergone boundary changes (mergers or splits) or contained missing values in any key variable were excluded to avoid inconsistencies and biases that might arise from inappropriate imputations (Meng, 1994). In addition, a small number of authorities (nine in total) were either governed by minor parties (Liberal Democrats) or experienced frequent changes of party control during the research period. These cases were also removed, as they could disproportionately influence the overall estimates. The final balanced panel consists of 116 local authorities (see Table 3).

The data preprocessing stage involved three standardization steps. First, temporal alignment: All variables were adjusted to match the UK fiscal year, which begins on April 1. Second, inflation adjustment: All monetary variables were deflated to constant 2015 prices using the Consumer Prices Index including owner occupiers' housing costs (Office for National Statistics, 2025b). Third, variable scaling: All continuous variables were pooled and standardized to a z-score before model estimation. For ease of interpretation, the final results will be transformed back to the original scale.

Table 3. Descriptive Statistics

Variable Name	Variable Category	N	Mean	Median	SD	Q10	Q90
Overall	Dependent	928	2.5	2.5	0.3	2.1	2.8
LATE_P18	Mediating	928	526.9	510.5	92.0	420.5	654.3
Unemp_Rate	Control	928	46.3	41.7	25.5	17.5	83.4
Mean	Control	928	28052.8	26162.0	6317.3	22683.8	35307.9
P_Density	Control	928	2451.5	1344.9	2735.2	228.4	5748.9

Finally, this study relies exclusively on publicly available secondary datasets that do not contain individual-level or sensitive personal information. All data were accessed through official government and regulatory sources, ensuring compliance with ethical standards and data protection regulations. As such, the research poses no risks to human participants and does not require additional ethical approval.

3.2 Empirical Strategy and Estimation

To examine whether changes in the political composition of local authorities alter the growth trajectory of care home quality, and whether this operates through the pathway of local government preventive service expenditure, this study employs a parallel process latent growth model (PPLGM) combined with a Bayesian framework for explicitly modeling slope-level mediation effects (Duncan et al., 2013; Webb and Bywaters, 2024). This approach has the advantage of simultaneously capturing the dynamic evolution of multiple time-series processes while situating their interrelationships within a unified modelling structure. As Cheong, MacKinnon and Khoo (2003) note, PPLGM can be used to assess how an independent variable influences the growth of a mediator, which in turn affects the growth of an outcome, by parallel modelling the trajectories of the mediator and outcome, enabling precise mediation testing.

In this study, the political composition and its changes, local government preventive expenditure, and the development of care home quality are all conceptualized as time-varying processes. PPLGM enables these trajectories to be estimated in parallel, allowing for the testing of whether the expenditure slope mediates the effect of politics on the trajectory of quality improvement. The use of Bayesian estimation further strengthens the approach. Unlike traditional frequentist methods that rely on asymptotic normality, the Bayesian framework incorporates prior knowledge, generates posterior distributions and credible intervals, and facilitates more transparent interpretation. This is particularly advantageous in this study, where the sample size is limited (see Table A1) (van de Schoot et al., 2014; Hespanhol et al., 2019; Smid et al., 2020).

This study divides the sample of local authorities into two mutually exclusive subsamples. The first consists of authorities with stable long-term party control ($g_i^A \in \{LAB, CON, NOC\}$) The second consists of authorities experiencing net party shifts ($g_i^B \in \{LAB \to CON, LAB \to NOC, CON \to LAB, CON \to NOC, NOC \to LAB, NOC \to CON\}$). The panel covers 2016–2023, with discrete time points t, centered as $t_c = t - \bar{t} \in$. This centering stabilizes slope estimates and facilitates cross-group comparison.

At the observational level, two time series are modelled in parallel: care home quality (Overall) and preventive expenditure per adult (LATE_P18). To account for contemporaneous shocks, two time-varying covariates are included: a pandemic pulse in 2020 $pulse_t = 1[year_t = 2020]$, and an exponential recovery term $rec_t^{(\cdot)} = 1 - \exp(-p_{(\cdot)} \cdot recovery_t)$, where $recovery_t = \max(0, year_t - 2020)$, and $p_{(\cdot)} \sim HalfNormal(1)$ governs recovery speed. On the z-scale, the measurement equations are:

$$Y_{it}^{over(z)} \sim \mathcal{N}(\mu_{it}^{over}, \sigma_{over}) \qquad \mu_{it}^{over} = \alpha_i^{over} + \beta_i^{over} t_c + \kappa_p^{over} pulse_t + \kappa_r^{over} rec_t^{over}$$

$$Y_{it}^{late(z)} \sim \mathcal{N}(\mu_{it}^{late}, \sigma_{late}) \qquad \mu_{it}^{late} = \alpha_i^{late} + \beta_i^{late} t_c + \kappa_p^{late} pulse_t + \kappa_r^{late} rec_t^{late}$$

Where $\alpha_i^{(\cdot)}$and $\beta_i^{(\cdot)}$ represent the intercept and slope at the local authority level. To improve posterior geometry and sampling efficiency, both parameters are specified using non-centred hierarchical parameterization:

$$\alpha_i^{(\cdot)} = \mu_{\alpha,(\cdot)} + z_i^{(\alpha)} \sigma_{\alpha,(\cdot)} \qquad \beta_{base,i}^{(\cdot)} = \mu_{\beta,(\cdot)} + z_i^{(\beta)} \sigma_{\beta,(\cdot)} \qquad z_i^{(\alpha)}, z_i^{(\beta)} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$$

Within this framework, party stability (Group A) or party change (Group B) enters the slope level to influence quality trajectories. At the same time, the growth of preventive expenditure per capita is modelled in parallel. Structural controls are introduced at the slope level, and the trajectory of preventive expenditure is allowed to affect the trajectory of care home quality through a coupling coefficient b. One category serves as the identification reference. All slope parameters are expressed in z-scale units of "standard deviations per year." For instance, $\theta_{over}^A[CON]$ can be interpreted as "relative to the CON baseline, the net annual change (in z-units per year) for other parties".

For Group A, the specification is as follows (for Group B, replace θ^A with θ^B):

$$\beta_i^{late} = \beta_{base,i}^{late} + \theta_{late}^A[g_i^A] + C_i^{(z)\intercal}\eta \qquad \theta_{late}^A[CON] = 0$$

$$\beta_i^{over} = \beta_{base,i}^{over} + b\beta_i^{late} + \theta_{over}^A[g_i^A] + C_i^{(z)\intercal}\delta \qquad \theta_{late}^A[CON] = 0$$

Where $C_i^{(z)} = \left[Mean_i^{(z)}, Unemp\_Rate_i^{(z)}, P\_Density_i^{(z)}\right]^\intercal$ denotes the local authority means of three structural covariates (each standardized annually within 2016−2023 and then averaged over time). Parameters η and δ captures the influence of these controls on the annual slopes of preventive expenditure and care home quality, respectively. The parameter b quantifies the average increment in the quality trajectory for each unit increase in the expenditure trajectory. Pandemic shocks and exponential recovery are modelled with separate coefficients and recovery rates (κ_p, κ_r, ρ) across the two processes, allowing for differential sensitivity to external shocks. All notation and variable definitions are summarized in Table A1.

3.3 Estimation and Inference

This study employed the NUTS sampler in NumPyro for Bayesian estimation (Homan and Gelman, 2014; Bradbury et al., 2018). All models were run with five chains, each with 3,000 warm-up iterations and 2,000 post–warm-up draws, yielding a total of 10,000 samples. The target acceptance rate was set to 0.99, and 64-bit floating-point precision was enabled to improve numerical stability in hierarchical structures. Convergence was assessed using no divergences, $\hat{R}$ values close to 1, and the effective sample size. Pointwise log−likelihoods were extracted during model fitting, and both LOO and a Bayesian Information Criterion (BIC−like) index (based on posterior mean log−likelihoods of $Y^{Overall}$) were computed to ensure that conclusions is not depend on a specific trend specification (Schwarz, 1978; Gelman et al., 1995; Watanabe and Opper, 2010; Vehtari et al., 2017). It is worth noting that ELPD LOO emphasizes predictive accuracy, while BIC-like criteria penalize additional parameters more heavily, prioritizing parsimony. Using both provides complementary perspectives: one focuses on predictive performance, while the other focuses on model simplicity.

Parameter inference and reporting followed Bayesian reporting standards. Following Webb and Bywaters (2024), this study present posterior means and 89% highest density intervals (HDIs) for key parameters, along with the probability of direction (PD) (McElreath, 2018). To emphasize substantive significance, a Region of Practical Equivalence (ROPE) was defined using 10% of the sample standard deviation as the threshold. Report the proportion of posterior mass falling below, within, and above the ROPE (Kruschke, 2014). Mediation and total effects were estimated by multiplying draws at each iteration, thereby fully propagating uncertainty (Tofighi and Kelley, 2020).

3.4 Robustness and Sensitivity Analysis

To assess the sensitivity of the main conclusions to sample partitioning and the strength of regularization, this study further estimated a joint hierarchical model. This specification simultaneously identifies the slope effects of long-term party control (Group A) and net party shifts (Group B) within a single framework. Let G_A and G_B denote the design matrices, and let, C_i represent the vector of time–averaged control variables for the local authority i. The slopes of per capita social care spending and care home quality were specified as follows:

$$\beta_i^{late} = \beta_{0i}^{late} + G_{A,i}\theta_A^{late} + G_{B,i}\theta_B^{late} + C_i\eta$$

$$\beta_i^{over} = \beta_{0i}^{over} + b\beta_i^{late} + G_{A,i}\emptyset_A^{over} + G_{B,i}\emptyset_B^{over} + C_i\delta$$

At the observational level, the centred time variable t_c was augmented with a 2020 pulse and an exponential recovery term:

$$\mu_{it}^{(\cdot)} = \alpha_i^{(\cdot)} + \beta_i^{(\cdot)}t_c + \kappa_{pulse}^{(\cdot)} \cdot 1\{t = 2020\} + \kappa_{rec}^{(\cdot)}(1 - e^{-\rho^{(\cdot)}recovery_t})$$

Where $recovery_t \in \{0,1,2,3\}$ indicates the number of years since 2020. Group A used CON as the baseline, while Group B used LAB–NOC as the baseline to ensure identification; all group effects are interpreted relative to these baselines. This joint specification adopted the same sampling setup, convergence diagnostics, out-of-sample fit evaluation, and reporting standards as the main models.

The sensitivity analysis proceeded along three main dimensions. First, functional form: for five time-varying variables (Overall, LATE_P18, Unemp_Rate, Mean, Pop_Density), this study fitted both linear and quadratic latent growth functions and compared model performance using LOOIC and BIC. Posterior predictive checks and graphical inspection informed the subsequent model specification and inference. Second, sample pooling: compared slope effects for care home quality from the joint model against those from the full split-sample models, using identical reporting metrics. This study focused on the consistency of posterior means in direction, the overlap of 89% intervals, and the stability of pd values. Third, prior sensitivity: examined whether conclusions changed under stronger versus weaker regularization of group effects. If the results remained consistent across sample pooling, prior tightness, functional form, and pandemic dynamics, this would indicate that the inferences regarding the effects of long-term party control and party shifts on quality growth slopes are robust.

Chapter 4: Findings

This chapter presents the empirical results of the model, focusing on three aspects: the choice of growth functions, partisan effects, and model robustness. First, linear and quadratic trends are compared to evaluate the appropriateness of different growth functions and to identify the optimal specification. Second, the analysis examines how long-term partisan structures and partisan shifts influence the trajectory of nursing home quality, while also testing the potential mediating role of per capita adult social care spending. Finally, joint modelling is employed to assess the robustness of the main findings under alternative specifications.

4.1 Growth-function Selection

Table 4 reports the comparison between linear and quadratic growth specifications. Overall, leave-one-out cross-validation (ELPD LOO) suggests that allowing curvature through a quadratic trend improves model fit for most variables. Specifically, for care quality (Overall) and per capita adult social care spending (LATE_P18), the quadratic model improves the ELPD by 9 and 26, respectively. These gains are modest but consistently favour the inclusion of a quadratic term. For the unemployment rate (Unemp_Rate) and population density (P_Density), the improvements are more substantial (123 and 144), indicating pronounced nonlinear dynamics in these cases. Although the improvement for population density is large, the high standard error (1035) indicates considerable uncertainty, likely driven by skewed distributions or the influence of extreme cases. In contrast, for the mean income (Mean), the quadratic model performs worse, with the ELPD decreasing by approximately 151, suggesting that additional curvature introduces noise rather than improving the fit.

From the perspective of information criteria, BIC-like results strongly favour the linear specification in nearly all cases. Except for mean income, quadratic models perform substantially worse, with differences ranging from 1,000 to 1,600, reflecting the heavy penalty imposed by additional parameters. For mean income, however, the quadratic model achieves a BIC-like score about 339 lower than the linear model, reinforcing the conclusion that a linear trend is more appropriate for this variable.

Taken together, only in the case of unemployment does the quadratic model achieve an ELPD/S.E. ratio above 2, providing statistically meaningful evidence of improvement. For all other variables, the ratios fall well below this threshold, indicating insufficient support for a systematic advantage of quadratic trends. Combined with the strong BIC-like preference for linear specifications, the results suggest that the benefits of quadratic models are limited to specific contexts. At the same time, their complexity penalties generally outweigh potential gains. Overall, the linear model offers a more balanced trade-off between predictive performance and parsimony. Consequently, the subsequent analysis proceeds with the linear specification.

Table 4. Model Fit Comparison

Variable	Model	ELPD LOO	S.E.	BIC	ELPD diff.	S.E. diff.	BIC diff.	ELPD/S.E. ratio
Overall	Linear	−47.17	23.72	3065.69				
	Quadratic	−38.12	22.69	4632.86	9.05	32.86	1567.17	0.28
LATE_P18	Linear	−4842.66	88.21	12289.53				
	Quadratic	−4816.55	75.30	13619.68	26.11	116.20	1330.15	0.22
Mean	Linear	−28497.61	3538.00	54300.63				
	Quadratic	−28648.75	3110.82	53961.45	−151.14	4688.40	-339.18	−0.03
Unemp_Rate	Linear	−4078.07	40.60	11098.80				
	Quadratic	−3954.86	37.15	12433.10	123.21	55.29	1334.30	2.23
P_Density	Linear	−7254.80	779.71	16782.53				
	Quadratic	−7110.71	678.53	17814.46	144.09	1034.80	1031.93	0.14

Note. S.E. Equal ELPD LOO Standard Error

4.2 Political Control and Outcome-trend Effects

The results for long-term party structure (Group A) suggest that, between 2016 and 2023, local authorities governed predominantly by Labour (LAB) or with No Overall Control (NOC) exhibited, on average, slightly more substantial improvements in service quality relative to those under Conservative control. Specifically, the estimated mean effects for Labour-led and NOC local authorities were 0.08 and 0.07, respectively. However, the associated HDPI 89% span across zero (LAB: -0.04 to 0.20; NOC: -0.03 to 0.18), indicating that the apparent advantages are not statistically robust and could plausibly be attributed to random variation. In probabilistic terms, the posterior distributions exhibit a relatively balanced allocation across the region of practical equivalence (ROPE), further underscoring the lack of decisive evidence for a systematic party-political effect (see Table 5, Figure 2 & Figure 3).

Table 5. Results of the Long-Term Party Structure Model

	Mean (z)	HDI 89% [low, high]	PD (%)	ROPE Below	ROPE Within	ROPE Above
Residual variance (σ²)	0.77	[0.73, 0.81]	1.00	0.00	0.00	1.00
Labour	0.08	[-0.04, 0.20]	0.86	0.01	0.60	0.39
No overall control	0.07	[-0.03, 0.18]	0.86	0.01	0.67	0.32
LATE_P18	-0.01	[-0.41, 0.41]	0.51	0.36	0.31	0.33
Mean	0.00	[-0.06, 0.06]	0.52	0.00	0.99	0.00
Unemp_Rate	0.04	[-0.04, 0.14]	0.78	0.01	0.84	0.15
P_Density	0.00	[-0.05, 0.05]	0.55	0.00	1.00	0.00
Pulse	-1.65	[-1.84, -1.44]	1.00	1.00	0.00	0.00
Recovery Amplitude	-1.06	[-1.132, -0.80]	1.00	1.00	0.00	0.00
Recovery Rate	2.53	[1.72, 3.31]	1.00	0.00	0.00	1.00

Note. Ref Conservative Party

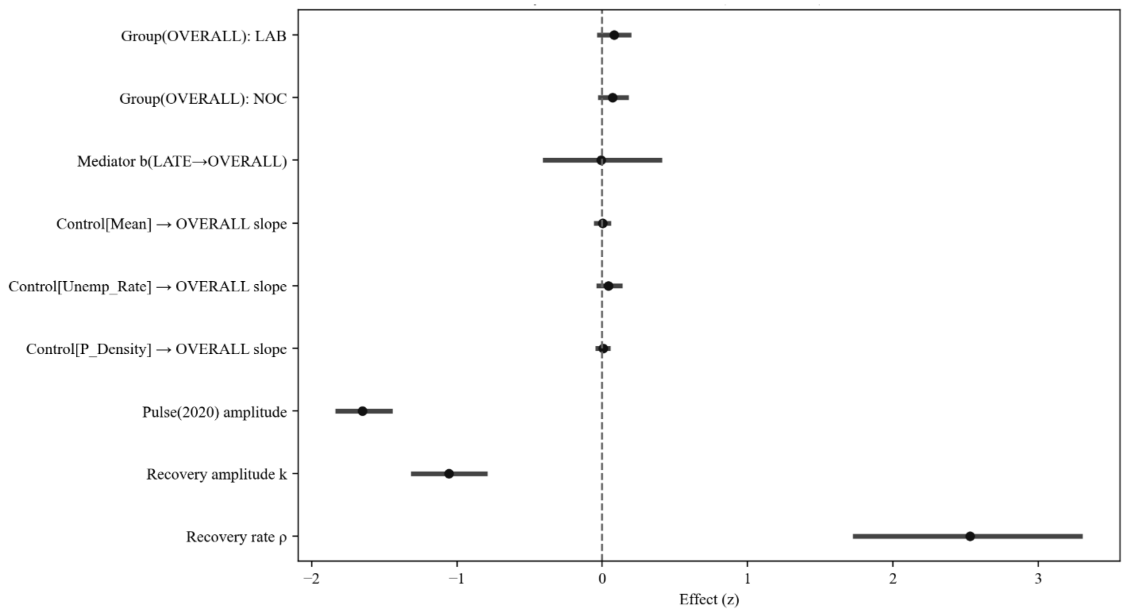

Figure 2. Bayesian Coefficient Forest Plot of Long-term Party Structure on Care Home Quality

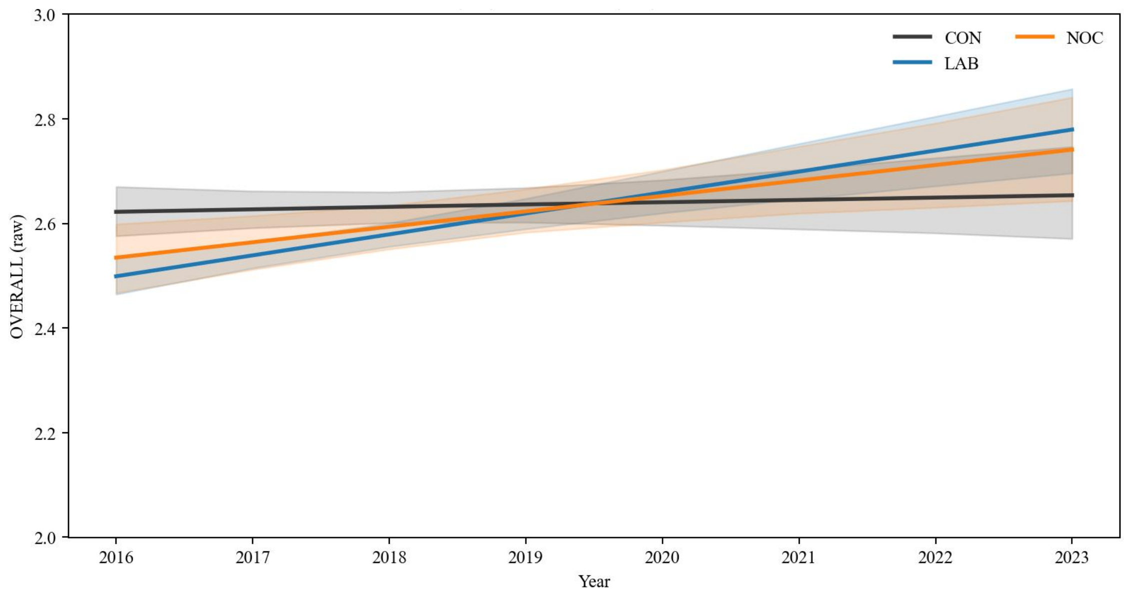

Figure 3. Long-term Party Structure Trends on Care Home Quality

Turning to the mediation pathway, the slope coefficient linking expenditure to quality outcomes is close to zero (-0.01). This near-null result suggests that, within this model specification, local authority spending does not act as a meaningful amplifier or suppressor of the observed trajectories of service quality. Put differently, differences in expenditure levels or their rate of change do not appear to condition the relationship between party control and quality outcomes. Moreover, the expenditure equation itself reveals minimal party-based differences, and the effects of key structural covariates (such as unemployment rate and population density) on the expenditure slope remain weak(see Figure B1). As a result, the model does not generate a stable or substantively significant net effect via the mediation channel.

These findings suggest that **H1** (that long-term party structure influences care home quality) is not supported. Although there are observable differences in the estimated quality slopes across local authorities with different governing arrangements, these differences are minor in magnitude and statistically inconclusive. Similarly, **H2** (that long-term party structure affects care home quality through preventive expenditure) is not supported under the present specification. The near-zero expenditure slope suggests that spending patterns are neither systematically shaped by long-term party structure nor substantially transmitted to quality outcomes at the slope level. In sum, both the direct and mediated pathways provide limited evidence that partisan control is a decisive driver of care home quality over the long run.

Second, with respect to party changes (Group B), the model shows that, relative to the baseline Labour to No Overall Control (LAB→NOC), the change from Conservative to Labour (CON→LAB) is associated with a significantly steeper improvement slope (Mean: 0.37, HDI 89%: 0.08/0.64). Moreover, 0.94 of the posterior mass lies outside the ROPE (±0.1) on the positive side (see Table 6, Figures 4 and 5). This indicates that a transition from a Conservative to a Labour government is followed by a significantly faster annual rate of improvement in care home quality. In contrast, other directions of party change—such as Conservative to No Overall Control (CON→NOC), Labour to Conservative (LAB→CON), or shifts involving NOC towards either major party exhibit either slight negative or near-zero effects. Their HDI 89% intervals consistently straddle zero, and their posterior mass is primarily contained within or around the ROPE, offering little support for the notion of systematic improvement or deterioration.

Table 6. Results of the Political Alternation Model

	Mean (z)	HDI 89% [low, high]	PD (%)	ROPE		
				Below	Within	Above
Residual variance (σ^2)	0.67	[0.61, 0.72]	1.00	0.00	0.00	1.00
CON->LAB	0.37	[0.08, 0.64]	0.98	0.00	0.06	0.94
CON->NOC	-0.06	[-0.27, 0.13]	0.71	0.39	0.52	0.09
LAB->CON	-0.01	[-0.19, 0.17]	0.55	0.22	0.63	0.16
NOC->CON	-0.08	[-0.22, 0.06]	0.82	0.40	0.57	0.02
NOC->LAB	-0.12	[-0.31, 0.08]	0.84	0.56	0.40	0.04
LATE_P18	0.19	[-0.54, 0.90]	0.67	0.25	0.16	0.59
Mean	-0.05	[-0.19, 0.09]	0.72	0.29	0.67	0.04
Unemp_Rate	0.06	[-0.10, 0.22]	0.73	0.06	0.59	0.35
P_Density	0.01	[-0.15, 0.16]	0.53	0.12	0.72	0.16
Pulse	-1.27	[-1.54, -1.02]	1.00	1.00	0.00	0.00
Recovery Amplitude	-0.69	[-1.06, -0.33]	1.00	0.99	0.01	0.00
Recovery Rate	1.94	[0.97, 2.98]	1.00	0.00	0.00	1.00

Note. Ref Conservative Party to No Overall Control Party

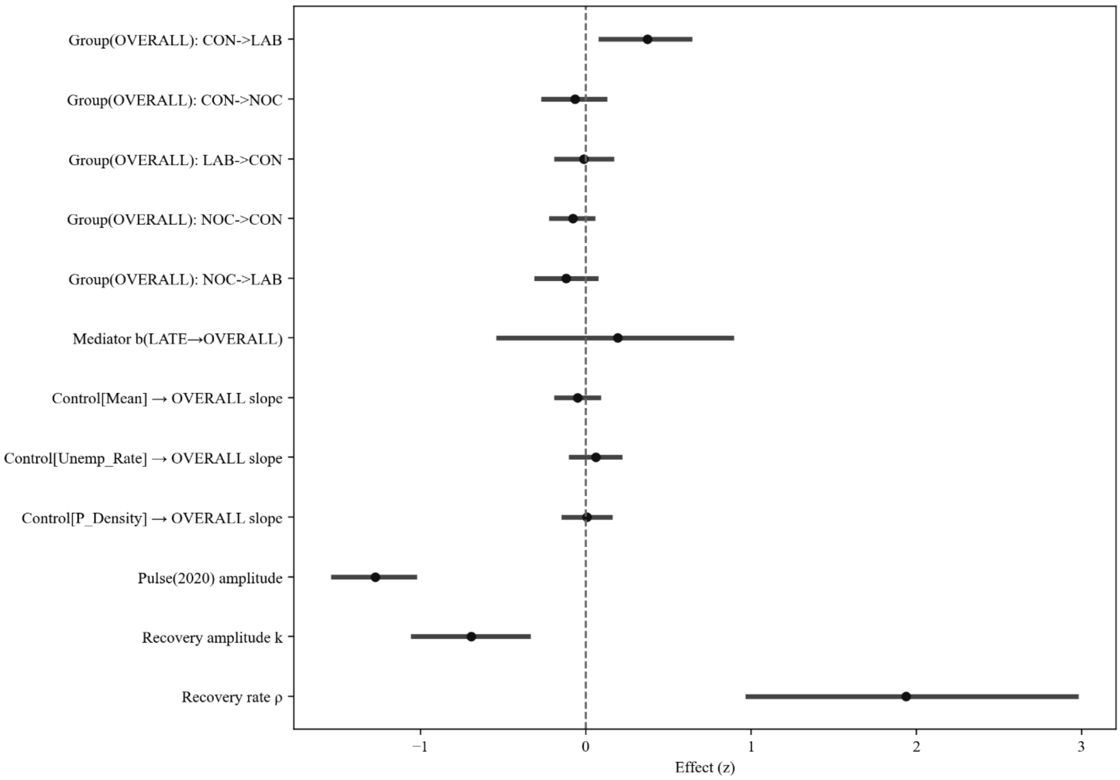

Figure 4. Bayesian Coefficient Forest Plot of Political Alternation on Care Home Quality

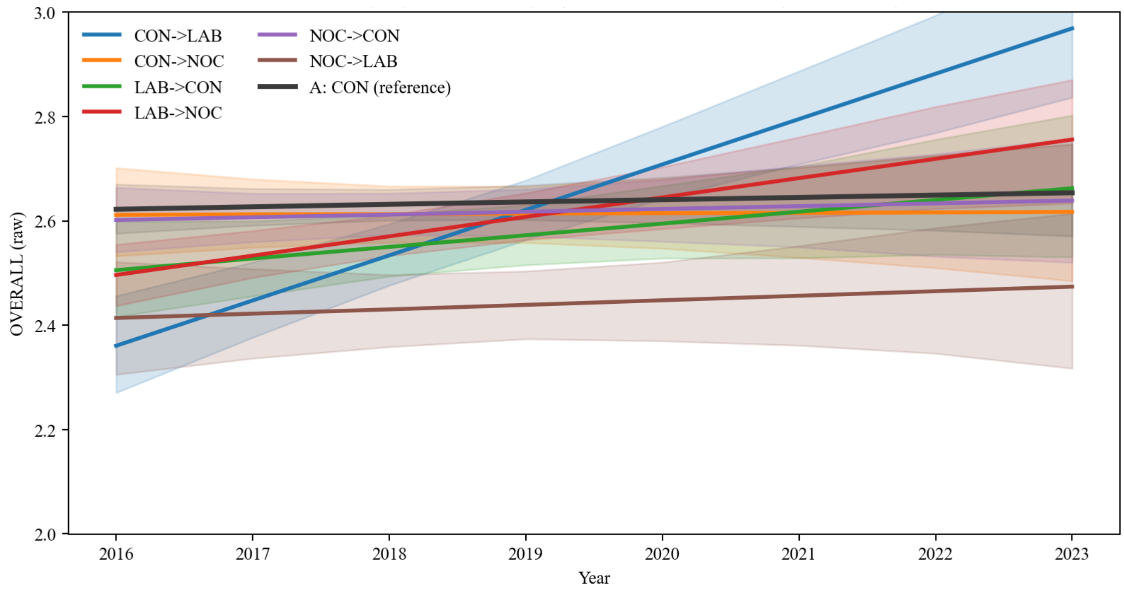

Figure 5. Political Alternation Trends on Care Home Quality

At the slope level, the estimated mediation pathway is also mildly positive but highly uncertain (Mean: 0.19, HDI 89%: -0.54/0.90). The wide credible interval indicates that the slope of per capita adult social care expenditure does not exert a stable or consistent amplifying effect on care home quality improvements. This means that while political turnover, particularly from Conservative to Labour, does correspond to measurable differences in quality trajectories, these differences cannot be robustly attributed to changes in expenditure patterns.

These findings suggest preliminary support for **H3** (party shifts influence care home quality): the transition from Conservative to Labour is associated with positive improvements, indicating that the "party change" pathway also plays a role. By contrast, **H4** (party shifts indirectly affect care home quality through preventive expenditure) is also not supported under the current specification and sample. The expenditure slope is neither consistently shaped by party shifts nor significantly transmitted to outcomes at the slope level.

Additionally, the pandemic-related parameters in all models align with expectations (see Figure 2 & Figure 4). The one-off shock in 2020 is strongly negative, the recovery effect is also adverse, and the recovery rate is relatively fast, indicating a "sharp decline followed by a rapid but incomplete rebound".

4.3 Model Diagnostics

To assess the robustness of the main findings across different model specifications, additional tests were conducted using the pooled sample. The results show that the previously positive association observed under long-term Labour control reverses slightly in the pooled model (Mean: -0.01), but the HDI 89% interval (-0.08/0.07) still spans zero (see Table 7 and Figure 6). Similarly, No Overall Control appears modest (Mean: 0.01), with its interval also crossing zero (HDI 89%: -0.08/0.12). This suggests that, under the joint modelling framework, the net effect of long-term party structure on the slope of care home quality remains limited.

Table 7. Results of the Joint Model

	Mean (z)	HDI 89% [low, high]	pd (%)	ROPE		
				Below	Within	Above
Residual variance (σ^2)	0.73	[0.70, 0.76]	1.00	0.00	0.00	1.00
Labour	-0.01	[-0.08, 0.07]	0.57	0.03	0.96	0.01
No overall control	0.01	[-0.08, 0.12]	0.58	0.03	0.90	0.08
CON->LAB	0.34	[0.19, 0.51]	1.00	0.00	0.01	0.99
CON->NOC	0.02	[-0.11, 0.14]	0.60	0.07	0.78	0.16
LAB->CON	0.06	[-0.09, 0.19]	0.74	0.03	0.66	0.31
NOC->CON	0.00	[-0.10, 0.10]	0.51	0.04	0.90	0.05
NOC->LAB	-0.02	[-0.19, 0.13]	0.59	0.22	0.67	0.11
LATE_P18	0.04	[-0.32, 0.39]	0.57	0.27	0.35	0.39
Mean	-0.02	[-0.08, 0.03]	0.74	0.01	0.99	0.00
Unemp_Rate	0.08	[0.01, 0.15]	0.97	0.00	0.68	0.32
P_Density	0.01	[-0.04, 0.06]	0.65	0.00	1.00	0.00
Pulse	-1.62	[-1.77, -1.46]	1.00	1.00	0.00	0.00
Recovery Amplitude	-1.06	[-1.26, -0.84]	1.00	1.00	0.00	0.00
Recovery Rate	2.75	[1.95, 3.52]	1.00	0.00	0.00	1.00

Note. Ref Conservative Party

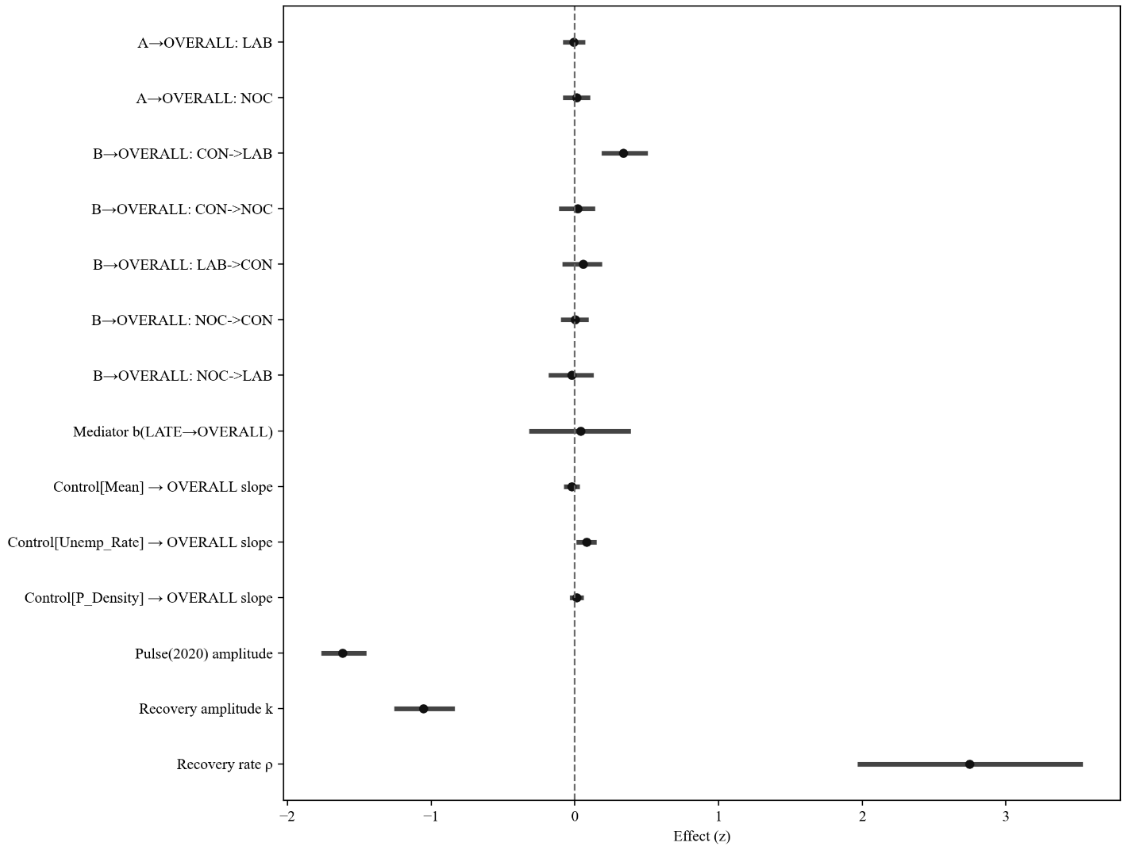

Figure 6. Bayesian Coefficient Forest Plot of Long-Term Party Structure and Political Alternation on Care Home Quality

By contrast, the results for party shifts confirm that the Conservative-to-Labour transition continues to exhibit a significantly positive slope difference (see Table 7 and Figure 6). Its HDI 89% does not cross zero, and most of the posterior mass lies outside the ROPE (±0.1 z/year) on the positive side. This indicates that, even under the stricter conditions of joint estimation, a "Conservative-to-Labour" transition remains associated with a faster rate of improvement in care home quality. In comparison, the point estimates for other types of party change are generally small, with intervals typically crossing zero, providing no consistent evidence of systematic differences.

At the slope level, the mediation parameter for per capita adult social care expenditure is also centred around zero (0.04) with a wide interval (-0.32/0.39), indicating that the expenditure slope does not provide a stable or significant net transmission to the quality slope. The pandemic-related parameters display the same directional patterns as in the main models.

Moreover, no divergent transitions occurred during sampling, and the diagnostic statistics indicate good convergence. All $\hat{R}$ values are below 1.01, and the effective sample sizes are generally high (bulk ESS exceeding 600 and tail ESS above 1500), further ensuring the reliability and robustness of the parameter estimates (see Table A3).

Chapter 5: Discussion

This chapter provides a discussion of the empirical findings and develops the analysis on both theoretical and policy fronts. First, it examines the role of party turnover in improving nursing home quality, highlighting its institutional significance as a critical political event. Second, it explores the interplay between long-term partisan control, economic conditions, and assesses their explanatory limitations in accounting for variations in quality. Finally, it reconsiders the role of fiscal expenditure as a channel for shaping public service performance, reflecting on both its mechanisms and constraints. Taken together, these discussions aim to place the research within the broader framework of public policy and governance, thereby clarifying its theoretical contributions and practical implications.

5.1 The Role of Political Transition

The most robust finding of this study is that the transition of power from the Conservative Party to the Labour Party exerted a significant and consistent positive effect on nursing home quality. This suggests that party turnover itself, as an independent political event, may serve as a critical driver of changes in care quality. The result resonates with institutionalist accounts of “critical junctures” (Collier and Collier, 2015) and “path dependence” (Pierson, 2000). Confirming that party ideology and priority-setting can shape the provision and performance of public services through path-dependent mechanisms (Imbeau et al., 2001). It further reinforces the long-observed differences between left- and right-wing parties in their approach to social welfare policy (Fernández-I-Marín et al., 2024); newly elected governments, facing heightened pressures of political accountability, often seek to advance their policy priorities, which leads to shifts in policy orientation and divergent outcomes (Hall, 1993; Häusermann et al., 2013). Such “administrative shocks” generated by political alternation may exert an influence that extends beyond contemporaneous economic fluctuations, thereby providing a critical window for substantive improvements in the quality of social services.

5.3 Interactions between Party Control and Economic Factors

Under long-standing partisan structures, local authorities governed by different parties exhibit varying trends in care home quality improvement, although these differences are not statistically significant. Among them, authorities long controlled by the Labour Party or with no overall party control show a preliminary association with faster improvements. This does not necessarily refute the view that parties influence public service quality. Instead, it points to a more important argument: the role of "political inertia" (Michels et al., 1915). Extended one-party dominance indirectly signals regional stability. Stable economic growth tends to translate into internal governmental incentives that favour the status quo, thereby reducing the likelihood of policy change (Ritchie, 2014; Zantvoort, 2017). Given that Labour has historically governed more industrialized urban areas, these regions face greater socioeconomic challenges but also often experience more rapid economic growth (Furlong, 2019; Claassen et al., 2024). In this context, financial factors may play a more central role in driving improvements in care quality. Of course, one cannot entirely rule out the possibility that such growth is itself partly an indirect outcome of partisan governance. However, this study is insufficient to substantiate this hypothesis.

5.2 Re-evaluating the Spending Pathway

This study further examined the mediating role of adult social care expenditure in the relationship between party control, party alternation, and nursing home quality. The findings show that neither long-term partisan structures nor shifts in party control exerts its influence through the growth rate of per capita adult social care spending. This suggests that partisan effects on public service performance operate primarily through direct channels rather than fiscal mediation. While this may seem counterintuitive, it is not uncommon in public management research. It supports earlier findings that differences in financial inputs by parties do not necessarily drive variations in performance (Boyne et al., 2012). For policymakers, this provides an important warning: simply increasing per capita expenditure is unlikely to yield substantial improvements in nursing home quality. In other words, fiscal resources are a necessary but not sufficient condition. While investment can enhance care-related quality of life, its marginal returns diminish as spending rises (Longo et al., 2023). Policy improvement should therefore rely more on institutional design, governance principles, and implementation capacity. Governments should focus on enhancing the effectiveness of fiscal resources through institutional and policy arrangements, rather than relying solely on expenditure expansion.

Chapter 6: Conclusions

6.1 Research Summary

This study employs a balanced panel dataset covering 116 local authorities in England from fiscal years 2016 to 2023, to examine the dynamic effects of local political structures on the quality of care homes. By integrating multiple official data sources and applying a parallel latent growth model with Bayesian estimation, it systematically analyzes how prolonged single-party control and partisan turnover shape the evolution of public service performance.

The central finding is that a shift in local government from Conservative to Labour leadership is a consistent and significant predictor of faster improvements in care home quality. In contrast, the net effect of long-term partisan dominance on quality trends is ambiguous. Moreover, increases in per capita adult social care spending do not appear to mediate the relationship between politics and service quality.

These results carry both theoretical and practical implications. Theoretically, they resonate with institutionalist and public management debates on "critical junctures" and "political inertia," highlighting the role of local politics in balancing path dependence and policy adjustment. Practically, they caution policymakers that improving public service quality cannot rely solely on greater financial inputs. Instead, institutional design and governance innovation could be more essential for enhancing the efficiency of resource use.

6.2 Limitations and Future Research

Despite its contributions, this study also acknowledges some limitations. First, although we compared linear and quadratic specifications, model choice remains constrained by parameterization and indicator selection. Results suggest that the quadratic model offers improvements for specific variables (e.g., unemployment rate), yet the linear specification is preferred for overall robustness. This implies that the chosen modelling framework may influence the conclusions. Future research could employ more flexible growth curves or hierarchical nonlinear models to capture the complex dynamics of policy performance.

Second, while the CQC framework provides a theoretically comprehensive standard for assessing care quality, its practical limitations have drawn increasing criticism. These include reporting delays, outdated data, and limited transparency in evaluation procedures (Department of Health and Social Care, 2024). Such issues contribute to a "measurement mismatch," where actual service quality diverges from regulatory ratings. Relying solely on CQC data may therefore introduce bias. Future work could combine CQC assessments with alternative digital data sources, such as user-generated online reviews, to capture both official evaluations and public perceptions of care quality.

Third, the study's sample design presents inherent constraints. Although election data were collected for all English local authorities, the analysis focuses on upper-tier councils, as they hold direct responsibility for public health and adult social care (Local Government Association, 2025). This focus naturally limits both the sample size and partisan diversity. Given the finding that fiscal expenditure is not a decisive factor in care home quality, concentrating only on authorities with primary fiscal powers may overlook other critical political dimensions. Future research could extend the analysis to lower-tier councils, where party turnover is more frequent and the presence of the Liberal Democrats and "no overall control" authorities has increased (see Figure B5). Incorporating these data would yield a more diverse and balanced partisan sample, potentially revealing how political dynamics at different levels of government influence variations in service quality.

References

Allan, S. and Forder, J. (2015) The Determinants of Care Home Closure. *Health Economics*, 24 (S1): 132–145. doi:10.1002/hec.3149.

Alonso, J.M. and Andrews, R. (2020) Political Ideology and Social Services Contracting: Evidence from a Regression Discontinuity Design. *Public Administration Review*, 80 (5): 743–754. doi:10.1111/puar.13177.

Antwi, Y.A. and Bowblis, J.R. (2018) The Impact of Nurse Turnover on Quality of Care and Mortality in Nursing Homes: Evidence from the Great Recession. *American Journal of Health Economics*, 4 (2): 131–163. doi:10.1162/ajhe_a_00096.

Bach-Mortensen, A., Goodair, B. and Esposti, M.D. (2024) Involuntary closures of for-profit care homes in England by the Care Quality Commission. *The Lancet Healthy Longevity*, 5 (4): e297–e302. doi:10.1016/S2666-7568(24)00008-4.

Baldwin, R., Chenoweth, L., dela Rama, M., et al. (2017) Does size matter in aged care facilities? A literature review of the relationship between the number of facility beds and quality. *Health Care Management Review*, 42 (4): 315. doi:10.1097/HMR.0000000000000116.

Barron, D.N. and West, E. (2017) The quasi-market for adult residential care in the UK: Do for-profit, not-for-profit or public sector residential care and nursing homes provide better quality care? *Social Science & Medicine*, 179: 137–146. doi:10.1016/j.socscimed.2017.02.037.

Bliss, A. (2022) *The State of Care report: what you need to know | NHS Confederation*. Available at: https://www.nhsconfed.org/publications/state-care-report (Accessed: 27 August 2025).

Borge, L.-E., Falch, T. and Tovmo, P. (2008) Public sector efficiency: the roles of political and budgetary institutions, fiscal capacity, and democratic participation. *Public Choice*, 136 (3): 475–495. doi:10.1007/s11127-008-9309-7.

Boyne, G. and Dahya, J. (2002) Executive Succession and the Performance of Public Organizations. *Public Administration*, 80 (1): 179–200. doi:10.1111/1467-9299.00299.

Boyne, G.A., James, O., John, P., et al. (2012) Party Control, Party Competition and Public Service Performance. *British Journal of Political Science*, 42 (3): 641–660. doi:10.1017/S0007123411000482.

Bradbury, J., Frostig, R., Hawkins, P., et al. (2018) *JAX: composable transformations of*

Python+ NumPy programs.

Bradley, E.H., Canavan, M., Rogan, E., et al. (2016) Variation In Health Outcomes: The Role Of Spending On Social Services, Public Health, And Health Care, 2000–09. *Health Affairs*, 35 (5): 760–768. doi:10.1377/hlthaff.2015.0814.

Bradley, E.H., Elkins, B.R., Herrin, J., et al. (2011) Health and social services expenditures: associations with health outcomes. *BMJ Quality & Safety*, 20 (10): 826–831. doi:10.1136/bmjqs.2010.048363.

Bureau of Labor Statistics (2025) *Local Area Unemployment Statistics Estimation Methodology*. Available at: https://www.bls.gov/lau/laumthd.htm (Accessed: 16 August 2025).

Cairney, P. and St Denny, E. (2020) *Why Isn't Government Policy More Preventive?* Oxford University Press. doi:10.1093/oso/9780198793298.001.0001.

Cardona, C., Anand, N.S., Alfonso, Y.N., et al. (2021) County health outcomes linkage to county spending on social services, building infrastructure, and law and order. *SSM - Population Health*, 16: 100930. doi:10.1016/j.ssmph.2021.100930.

Care Quality Commission (2025) *Using CQC data - Care Quality Commission*. Available at: https://www.cqc.org.uk/about-us/transparency/using-cqc-data (Accessed: 12 August 2025).

Castle, N.G. and Anderson, R.A. (2011) Caregiver Staffing in Nursing Homes and Their Influence on Quality of Care: Using Dynamic Panel Estimation Methods. *Medical Care*, 49 (6): 545–552.

Chang, S.-J. and Cheng, M.-A. (2013) The impact of nursing quality on nursing home efficiency: evidence from Taiwan Chang and Ehsan Feroz, H. (ed.). *Review of Accounting and Finance*, 12 (4): 369–386. doi:10.1108/RAF-06-2012-0058.

Cheong, J., MacKinnon, D.P. and Khoo, S.T. (2003) Investigation of Mediational Processes Using Parallel Process Latent Growth Curve Modeling. *Structural Equation Modeling: A Multidisciplinary Journal*, 10 (2): 238–262. doi:10.1207/S15328007SEM1002_5.

Claassen, C., Carman, C. and Bankov, P. (2024) Is there a rural-urban political divide in Britain? *Journal of Elections, Public Opinion and Parties*, 0 (0): 1–29. doi:10.1080/17457289.2024.2393587.

Collier, R.B. and Collier, D. (2015) *Shaping the political arena: Critical junctures, the labor movement, and regime dynamics in Latin America*. Available at:

https://escholarship.org/uc/item/8qr1z7gc (Downloaded: 26 August 2025).

Comondore, V.R., Devereaux, P.J., Zhou, Q., et al. (2009) Quality of care in for-profit and not-for-profit nursing homes: systematic review and meta-analysis. *BMJ*, 339: b2732. doi:10.1136/bmj.b2732.

Cummins, I. (2018) The Impact of Austerity on Mental Health Service Provision: A UK Perspective. *International Journal of Environmental Research and Public Health*, 15 (6): 1145. doi:10.3390/ijerph15061145.

Dawkins, R.C. (2017) *The Politics of Local Privatization: Privatized Service Delivery and the Strength of Our Local Democracy - ProQuest*. Available at: https://www.proquest.com/docview/1958944787?pq-origsite=gscholar&fromopenview=true&sourcetype=Dissertations%20&%20Theses (Accessed: 28 July 2025).

Dellefield, M.E., Castle, N.G., McGilton, K.S., et al. (2015) The Relationship Between Registered Nurses and Nursing Home Quality: An Integrative Review (2008-2014). *Nursing Economic$*, 33 (2): 95–108, 116.

Department of Health and Social Care (2024) *Review into the operational effectiveness of the Care Quality Commission: interim report*. Available at: https://www.gov.uk/government/publications/review-into-the-operational-effectiveness-of-the-care-quality-commission/review-into-the-operational-effectiveness-of-the-care-quality-commission-interim-report (Accessed: 28 July 2025).

Di, M.D., Ferrara, A. and Mariotti, I. (2022) Place-Based Policies and Political Discontent: The Mediating Role of Local Spending Programmes. *SCIENZE REGIONALI*. doi:10.14650/105122.

Dintrans, P.V. (2018) Do long-term care services match population needs? A spatial analysis of nursing homes in Chile. *PLOS ONE*, 13 (6): e0199522. doi:10.1371/journal.pone.0199522.

Donabedian, A. (1988) The Quality of Care: How Can It Be Assessed? *JAMA*, 260 (12): 1743–1748. doi:10.1001/jama.1988.03410120089033.

Dormont, B. and Martin, C. (2012) "Quality of service and cost-efficiency of French nursing homes." In *9th European Conference on Health Economics (ECHE), Zurich*. 2012.

Dulal, R. (2018) Technical efficiency of nursing homes: do five-star quality ratings matter? *Health Care Management Science*, 21 (3): 393–400. doi:10.1007/s10729-017-9392-8.

Duncan, T.E., Duncan, S.C. and Strycker, L.A. (2013) *An Introduction to Latent Variable Growth Curve Modeling: Concepts, Issues, and Application, Second Edition*. 2nd edn. New York: Routledge. doi:10.4324/9780203879962.

Fernández-I-Marín, X., Knill, C. and Steinebach, Y. (2024) Do parties matter for policy accumulation? An analysis of social policy portfolios in 22 countries. *European Journal of Political Research*, 63 (3): 1175–1196. doi:10.1111/1475-6765.12642.

Fieldhouse, E. and Bailey, J. (2023) A new electoral map? Brexit, realignment and the changing geography of party support in England. *Political Geography*, 102: 102862. doi:10.1016/j.polgeo.2023.102862.

Forder, J. and Allan, S. (2014) The impact of competition on quality and prices in the English care homes market. *Journal of Health Economics*, 34: 73–83. doi:10.1016/j.jhealeco.2013.11.010.

Foster, D. and Harker, R. (2025) *Adult social care funding in England*. Available at: https://commonslibrary.parliament.uk/research-briefings/cbp-7903/ (Accessed: 28 August 2025).

Fung, C.H., Lim, Y.-W., Mattke, S., et al. (2008) Systematic Review: The Evidence That Publishing Patient Care Performance Data Improves Quality of Care. *Annals of Internal Medicine*, 148 (2): 111–123. doi:10.7326/0003-4819-148-2-200801150-00006.

Furlong, J. (2019) The changing electoral geography of England and Wales: Varieties of “left-behindedness.” *Political Geography*, 75: 102061. doi:10.1016/j.polgeo.2019.102061.

Gelman, A., Carlin, J.B., Stern, H.S., et al. (1995) *Bayesian Data Analysis*. New York: Chapman and Hall/CRC. doi:10.1201/9780429258411.

GOV UK (2025) *Local authority revenue expenditure and financing*. Available at: https://www.gov.uk/government/collections/local-authority-revenue-expenditure-and-financing (Accessed: 12 August 2025).

Grabowski, D.C. (2004) A Longitudinal Study of Medicaid Payment, Private-Pay Price and Nursing Home Quality. *International Journal of Health Care Finance and Economics*, 4 (1): 5–26.

Gray, M. and Barford, A. (2018) The depths of the cuts: the uneven geography of local government austerity. *Cambridge Journal of Regions, Economy and Society*, 11 (3): 541–563. doi:10.1093/cjres/rsy019.

Grembi, V., Nannicini, T. and Troiano, U. (2016) Do Fiscal Rules Matter? *American*

Economic Journal: Applied Economics, 8 (3): 1–30. doi:10.1257/app.20150076.

Hall, P.A. (1993) Policy Paradigms, Social Learning, and the State: The Case of Economic Policymaking in Britain. *Comparative Politics*, 25 (3): 275–296. doi:10.2307/422246.

Harrington, C., Stockton, J. and Hooper, S. (2014) The Effects of Regulation and Litigation on a Large for-Profit Nursing Home Chain. *Journal of Health Politics, Policy and Law*, 39 (4): 781–809. doi:10.1215/03616878-2743039.

Harrington, C., Woolhandler, S., Mullan, J., et al. (2001) Does Investor Ownership of Nursing Homes Compromise the Quality of Care? *American Journal of Public Health*, 91 (9): 1452–1455. doi:10.2105/AJPH.91.9.1452.

Häusermann, S., Picot, G. and Geering, D. (2013) Review Article: Rethinking Party Politics and the Welfare State – Recent Advances in the Literature. *British Journal of Political Science*, 43 (1): 221–240. doi:10.1017/S0007123412000336.

Hendriks, F., Lidström, A. and Loughlin, J. (eds) (2010) *The Oxford Handbook of Local and Regional Democracy in Europe*. Oxford University Press. doi:10.1093/oxfordhb/9780199562978.001.0001.

Hespanhol, L., Vallio, C.S., Costa, L.M., et al. (2019) Understanding and interpreting confidence and credible intervals around effect estimates. *Brazilian Journal of Physical Therapy*, 23 (4): 290–301. doi:10.1016/j.bjpt.2018.12.006.

Hick, R. (2022) Austerity, Localism, and the Possibility of Politics: Explaining Variation in Three Local Social Security Schemes Between Elected Councils in England. *Sociological Research Online*, 27 (2): 251–272. doi:10.1177/1360780421990668.

Hillmer, M.P., Wodchis, W.P., Gill, S.S., et al. (2005) Nursing Home Profit Status and Quality of Care: Is There Any Evidence of an Association? *Medical Care Research and Review*, 62 (2): 139–166. doi:10.1177/1077558704273769.

HM Government (2022) *Levelling Up the United Kingdom: executive summary*. Available at: https://www.gov.uk/government/publications/levelling-up-the-united-kingdom/levelling-up-the-united-kingdom-executive-summary (Accessed: 26 July 2025).

Homan, M.D. and Gelman, A. (2014) The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15 (1): 1593–1623.

Hudson, B. (2019) Commissioning for change: A new model for commissioning adult social care in England. *Critical Social Policy*, 39 (3): 413–433. doi:10.1177/0261018318818960.

Imbeau, L.M., Pétry, F. and Lamari, M. (2001) Left–right party ideology and government policies: A meta–analysis. *European Journal of Political Research*, 40 (1): 1–29. doi:10.1111/1475-6765.00587.

Iparraguirre, J.L. (2020) Reductions in local government spending on community-based social care and unmet social care needs of older people in England. *The Journal of the Economics of Ageing*, 17: 100126. doi:10.1016/j.jeoa.2017.07.001.

Iqbal, A., Kinghorn, P., Glasby, J., et al. (2023) A Scoping Review of the Costs, Consequences, and Wider Impacts of Residential Care Home Closures in a UK Context. *Health & Social Care in the Community*, 2023 (1): 8675499. doi:10.1155/2023/8675499.

Jacobs, A.M. (2011) *Governing for the Long Term: Democracy and the Politics of Investment*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511921766.

Knapp, M., Hardy, B. and Forder, J. (2001) Commissioning for Quality: Ten Years of Social Care Markets in England. *Journal of Social Policy*, 30 (2): 283–306. doi:10.1017/S0047279401006225.

Krieger, N., Moallef, S., Chen, J.T., et al. (2024) Politicians, power, and the people’s health: US elections and state health outcomes, 2012–2024. *Health Affairs Scholar*, 2 (12): qxae163. doi:10.1093/haschl/qxae163.

Kruschke, J. (2014) *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan*. San Diego, UNITED STATES: Elsevier Science & Technology. Available at: http://ebookcentral.proquest.com/lib/bristol/detail.action?docID=5754481 (Downloaded: 15 August 2025).

Laine, J., Linna, M., Häkkinen, U., et al. (2005) Measuring the productive efficiency and clinical quality of institutional long-term care for the elderly. *Health Economics*, 14 (3): 245–256. doi:10.1002/hec.926.

Liddle, J., Shutt, J. and Addidle, G. (2022) Editorial: Levelling up the United Kingdom? A useful mantra but too little substance or delivery? *Local Economy*, 37 (1–2): 3–12. doi:10.1177/02690942221099835.

Local Government Association (2024) *Must know guide: The annual budget process | Local Government Association*. Available at: https://www.local.gov.uk/publications/must-know-guide-annual-budget-process (Accessed: 15 August 2025).

Local Government Association (2025) *Councillors’ guide to local authority public health* *responsibilities | Local Government Association*. Available at: https://www2.local.gov.uk/publications/councillors-guide-local-authority-public-health-responsibilities-0 (Accessed: 26 August 2025).

Longo, F., Claxton, K., Lomas, J., et al. (2023) Is extending eligibility for adult social care better than investing more in existing users in England? A cross-sectional evidence for multiple financial years. *BMJ Open*, 13 (9): e070833. doi:10.1136/bmjopen-2022-070833.

McElreath, R. (2018) *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. New York: Chapman and Hall/CRC. doi:10.1201/9781315372495.

Meng, X.-L. (1994) Multiple-Imputation Inferences with Uncongenial Sources of Input. *Statistical Science*, 9 (4): 538–558.

Merritt, E.C. (2024) *Social care in England: Current situation, case for a strategy and further support for unpaid carers*. Available at: https://lordslibrary.parliament.uk/social-care-in-england-current-situation-case-for-a-strategy-and-further-support-for-unpaid-carers/ (Accessed: 27 August 2025).

Michels, R., Paul, E. and Paul, C. (1915) *Political parties; a sociological study of the oligarchical tendencies of modern democracy*. New York, Hearst's International Library Co. Available at: http://archive.org/details/politicalparties00mich (Downloaded: 26 August 2025).

Netten, A., Burge, P., Malley, J., et al. (2012) Outcomes of social care for adults: developing a preference weighted measure. *Health Technology Assessment*, 16 (16): 1–165. doi:10.3310/hta16160.

Netten, A., Darton, R. and Williams, J. (2003) Nursing home closures: effects on capacity and reasons for closure. *Age and Ageing*, 32 (3): 332–337. doi:10.1093/ageing/32.3.332.

Ng, S. (2021) *Modeling Macroeconomic Variations after Covid-19*. doi:10.3386/w29060.

Office for National Statistics (2024) *Earnings and hours worked, place of residence by local authority: ASHE Table 8 - Office for National Statistics*. Available at: https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/earningsandworkinghours/datasets/placeofresidencebylocalauthorityashetable8 (Accessed: 12 August 2025).

Office for National Statistics (2025a) *CC01 Regional labour market: Claimant Count by unitary and local authority (official statistics in development) - Office for National Statistics*. Available at:

https://www.ons.gov.uk/employmentandlabourmarket/peoplenotinwork/unemployment/datasets/claimantcountbyunitaryandlocalauthorityexperimental (Accessed: 12 August 2025).

Office for National Statistics (2025b) *Consumer price inflation tables - Office for National Statistics*. Available at: https://www.ons.gov.uk/economy/inflationandpriceindices/datasets/consumerpriceinflation (Accessed: 13 August 2025).

Office for National Statistics (2025c) *Open Geography Portal*. Available at: https://geoportal.statistics.gov.uk/search?q=BDY_CTYUA%20BFE%20december (Accessed: 12 August 2025).

Ogami, M. (2024) The Conditionality of Political Short-Termism: A Review of Empirical and Experimental Studies. *Politics and Governance*, 12 (0). doi:10.17645/pag.7764.

Ogden, K. and Phillips, D. (2024) *How have English councils' funding and spending changed? 2010 to 2024*. doi:10.1920/re.ifs.2024.0318.

Oronce, C.I.A., Ponce, N.A., Sarkisian, C.A., et al. (2025) Association between governmental spending on social services and health care use among low-income older adults. *Health Affairs Scholar*, 3 (1): qxae181. doi:10.1093/haschl/qxae181.

O'Toole, L.J., Jr. and Meier, K.J. (1999) Modeling the Impact of Public Management: Implications of Structural Context. *Journal of Public Administration Research and Theory*, 9 (4): 505–526. doi:10.1093/oxfordjournals.jpart.a024421.

Park, Y.J. and Martin, E.G. (2018) Geographic Disparities in Access to Nursing Home Services: Assessing Fiscal Stress and Quality of Care. *Health Services Research*, 53 (Suppl Suppl 1): 2932–2951. doi:10.1111/1475-6773.12801.

Pierson, P. (2000) Increasing Returns, Path Dependence, and the Study of Politics. *American Political Science Review*, 94 (2): 251–267. doi:10.2307/2586011.

Ritchie, F. (2014) *Resistance to change in government: Risk, inertia and incentives*. Available at: https://uwe-repository.worktribe.com/output/807308/resistance-to-change-in-government-risk-inertia-and-incentives (Accessed: 26 August 2025).

Rodden, J.A., Eskeland, G.S. and Litvack, J. (2003) *Fiscal Decentralization and the Challenge of Hard Budget Constraints*. The MIT Press. doi:10.7551/mitpress/3021.001.0001.

Sandford, M. (2025) *Local authority financial resilience*. Available at: https://commonslibrary.parliament.uk/research-briefings/cbp-8520/ (Accessed: 15

August 2025).

van de Schoot, R., Kaplan, D., Denissen, J., et al. (2014) A Gentle Introduction to Bayesian Analysis: Applications to Developmental Research. *Child Development*, 85 (3): 842–860. doi:10.1111/cdev.12169.

Schwarz, G. (1978) Estimating the Dimension of a Model. *The Annals of Statistics*, 6 (2): 461–464.

Smid, S.C., McNeish, D., Miočević, M., et al. (2020) Bayesian Versus Frequentist Estimation for Structural Equation Models in Small Sample Contexts: A Systematic Review. *Structural Equation Modeling: A Multidisciplinary Journal*, 27 (1): 131–161. doi:10.1080/10705511.2019.1577140.

Stevenson, D.G. (2006) Is a Public Reporting Approach Appropriate for Nursing Home Care? *Journal of Health Politics, Policy and Law*, 31 (4): 773–810. doi:10.1215/03616878-2006-003.

Stokes, J., Bower, P., Guthrie, B., et al. (2022) Cuts to local government spending, multimorbidity and health-related quality of life: A longitudinal ecological study in England. *The Lancet Regional Health - Europe*, 19: 100436. doi:10.1016/j.lanepe.2022.100436.

The Labour Party (2022) *A New Britain: Renewing our Democracy and Rebuilding our Economy*. Available at: https://labour.org.uk/updates/stories/a-new-britain-renewing-our-democracy-and-rebuilding-our-economy/ (Accessed: 26 July 2025).

Tofighi, D. and Kelley, K. (2020) Indirect Effects in Sequential Mediation Models: Evaluating Methods for Hypothesis Testing and Confidence Interval Formation. *Multivariate Behavioral Research*, 55 (2): 188–210. doi:10.1080/00273171.2019.1618545.

UK Parliament (2025) *Parliament and elections data*. Available at: https://commonslibrary.parliament.uk/data/parliament-elections-data/ (Accessed: 12 August 2025).

Vehtari, A., Gelman, A. and Gabry, J. (2017) Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27 (5): 1413–1432. doi:10.1007/s11222-016-9696-4.

Wang, Q., Ren, S. and Hou, Y. (2020) Atmospheric environmental regulation and industrial total factor productivity: the mediating effect of capital intensity. *Environmental Science and Pollution Research*, 27 (26): 33112–33126. doi:10.1007/s11356-020-

09523-4.

Watanabe, S. and Opper, M. (2010) Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of machine learning research*, 11 (12).

Webb, C. and Bywaters, P. (2024) Inequalities in out-of-home care rates in England: Does local party politics matter? *Child Abuse & Neglect*, 149: 106590. doi:10.1016/j.chiabu.2023.106590.

Werner, R.M., Norton, E.C., Konetzka, R.T., et al. (2012) Do consumers respond to publicly reported quality information? Evidence from nursing homes. *Journal of Health Economics*, 31 (1): 50–61. doi:10.1016/j.jhealeco.2012.01.001.

Yang, O., Yong, J. and Scott, A. (2022) Nursing Home Competition, Prices, and Quality: A Scoping Review and Policy Lessons. *The Gerontologist*, 62 (7): e384–e401. doi:10.1093/geront/gnab050.

Yi, C. and Qiu, X. (2025) From local government to local governance: A systematic literature review and future research agenda. *Australian Journal of Public Administration*, 84 (1): 102–124. doi:10.1111/1467-8500.12644.

Zantvoort, B. (2017) Political inertia and social acceleration. *Philosophy & Social Criticism*, 43 (7): 707–723. doi:10.1177/0191453716669195.

Zeng, J., Zhang, Y., Yin, Y., et al. (2024) The effect of the Covid pandemic on stock market volatility: Separating initial impact from time-to-recovery. *Data Science in Finance and Economics*, 4 (4): 531–547. doi:10.3934/DSFE.2024022.

Appendix A

Table A1. Number of Local Authorities Used in the Models in Group A and B

Category	Party Name	Number of Local Authorities
Group A	LAB	47
	CON	22
	NOC	10
Group B	LAB->NOC	11
	LAB->CON	4
	CON->NOC	5
	CON->LAB	4
	NOC->CON	10
	NOC->LAB	3

Table A2. Notation and Variable Mapping

Symbol	Definition	Description
i	Local authority index	/
t	Year Index (2016–2023)	Discrete time
t_c	Center period (2016–2023)	Centered to the mean to facilitate slope interpretation
Y_{it}^{over}, Y_{it}^{late}	Observed CQC Overall / LATE_P18 (z scale for modeling, back-transformed to original scale for display)	Standardized overall quality/spending metrics
$\alpha_i^{(\cdot)}$	Intercept (LA layer)	Baseline level of each LA
$\beta_i^{(\cdot)}$	Time slope (LA layer)	Annual trends for each LA
$\mu_{it}^{(\cdot)}$	Conditional mean	Mean for the normal distribution
$\sigma_{(\cdot)}$	Residual standard deviation	Observation layer noise
$pulse_t$	Epidemic pulse dummy variable (2020=1)	Sudden shock in 2020
$recovery_t$	Recovery time after 2020 (0/1/2/3)	Cumulative years after the epidemic
$rec_t^{(\cdot)} = (1 - e^{-\rho^{(\cdot)} recovery_t})$	Exponential recovery	Recovery dynamic process
$k_{pulse}^{(\cdot)}$	Epidemic pulse effect coefficient	Impact amplitude
$k_{rec}^{(\cdot)}$	Restoration amplitude coefficient	Recovery range
$\rho^{(\cdot)}$	Recovery speed parameters	Control recovery speed
$G_{A,i}$	Long-term party vector (one-hot)	/
$G_{B,i}$	Party change vector (one-hot)	/
θ^{over}	Relative contribution of each to the OVERALL slope (relative to the respective baseline)	/
θ^{late}	Relative contribution of groups to the LATE_P18 slope	/
b	Linkage coefficient from LATE_P18 slope to OVERALL slope	/
C_i	Structural controls for local authorities (vector of mean z scores)	Contains Mean, Unemp_Rate, P_Density
η	Effect of control variables on the expenditure slope	Long-term mean effect
δ	Effect of control variables on quality slope	Long-term mean effect

Table A3. MCMC Parameter Summary

Param	Mean	SD	HDI_3%	HDI_97%	MCSE_mean	MCSE_sd	ESS_bulk	ESS_tail	R_hat
alpha_late[0]	1.899	0.128	1.66	2.139	0.001	0.001	12669	6906	1
alpha_late[1]	1.977	0.128	1.74	2.218	0.001	0.001	11701	6436	1
alpha_late[2]	1.947	0.13	1.694	2.184	0.001	0.001	12858	7156	1
alpha_late[3]	-0.038	0.125	-0.275	0.195	0.001	0.001	21674	7275	1
alpha_late[4]	-0.212	0.127	-0.457	0.024	0.001	0.002	21631	6670	1
alpha_late[5]	2.05	0.126	1.82	2.291	0.001	0.001	11845	7606	1
alpha_late[6]	0.767	0.127	0.542	1.018	0.001	0.002	20664	7331	1
alpha_late[7]	1.185	0.127	0.951	1.429	0.001	0.002	17334	6911	1
alpha_late[8]	2.051	0.128	1.798	2.284	0.001	0.001	11869	6882	1
alpha_late[9]	-0.88	0.124	-1.107	-0.647	0.001	0.001	19674	7646	1
alpha_late[10]	-0.932	0.126	-1.177	-0.703	0.001	0.001	17449	7356	1
alpha_late[11]	-0.409	0.127	-0.648	-0.169	0.001	0.002	20566	6877	1
alpha_late[12]	-0.688	0.128	-0.93	-0.452	0.001	0.002	20225	6611	1
alpha_late[13]	0.554	0.13	0.308	0.797	0.001	0.002	20623	6339	1
alpha_late[14]	-0.179	0.125	-0.417	0.05	0.001	0.001	20062	7123	1
alpha_late[15]	0.478	0.128	0.224	0.708	0.001	0.002	22544	6716	1
alpha_late[16]	-0.148	0.123	-0.385	0.085	0.001	0.001	22325	6913	1
alpha_late[17]	-0.455	0.127	-0.695	-0.223	0.001	0.001	20925	6785	1
alpha_late[18]	0.607	0.124	0.371	0.839	0.001	0.001	18993	7383	1
alpha_late[19]	0.149	0.128	-0.092	0.388	0.001	0.002	21449	6691	1
alpha_late[20]	-0.016	0.128	-0.25	0.233	0.001	0.001	21026	6999	1
alpha_late[21]	-0.466	0.123	-0.693	-0.229	0.001	0.001	20807	7467	1
alpha_late[22]	-0.669	0.129	-0.915	-0.432	0.001	0.002	20489	6408	1
alpha_late[23]	-0.295	0.126	-0.535	-0.066	0.001	0.001	21179	7473	1
alpha_late[24]	-1.075	0.128	-1.316	-0.835	0.001	0.002	18382	6798	1
alpha_late[25]	-0.539	0.13	-0.782	-0.298	0.001	0.002	21834	6637	1
alpha_late[26]	0.577	0.126	0.335	0.808	0.001	0.001	19002	7392	1
alpha_late[27]	-0.986	0.126	-1.227	-0.755	0.001	0.002	18320	7037	1
alpha_late[28]	-0.709	0.128	-0.949	-0.468	0.001	0.002	20968	7280	1

alpha_late[29]	-1.249	0.129	-1.488	-1.003	0.001	0.002	18374	6320	1
alpha_late[30]	0.189	0.129	-0.048	0.432	0.001	0.002	20304	6721	1
alpha_late[31]	-0.741	0.129	-0.977	-0.495	0.001	0.002	18972	6572	1
alpha_late[32]	0.036	0.128	-0.204	0.277	0.001	0.001	21918	7328	1
alpha_late[33]	-0.324	0.126	-0.551	-0.081	0.001	0.001	21183	7103	1
alpha_late[34]	-0.29	0.128	-0.533	-0.048	0.001	0.002	21093	6689	1
alpha_late[35]	0.033	0.126	-0.203	0.267	0.001	0.001	20495	7222	1
alpha_late[36]	0.346	0.126	0.118	0.595	0.001	0.001	21492	7083	1
alpha_late[37]	-0.401	0.127	-0.644	-0.162	0.001	0.001	20051	7463	1
alpha_late[38]	0.015	0.126	-0.218	0.255	0.001	0.001	22745	7076	1
alpha_late[39]	-0.853	0.128	-1.093	-0.615	0.001	0.002	18335	7100	1
alpha_late[40]	0.02	0.128	-0.222	0.262	0.001	0.001	22932	7844	1
alpha_late[41]	0.035	0.128	-0.195	0.283	0.001	0.001	21874	7002	1
alpha_late[42]	0.714	0.125	0.481	0.952	0.001	0.001	18190	7292	1
alpha_late[43]	-0.509	0.127	-0.749	-0.274	0.001	0.001	22560	6898	1
alpha_late[44]	0.126	0.127	-0.119	0.353	0.001	0.002	19153	6816	1
alpha_late[45]	-0.563	0.13	-0.808	-0.325	0.001	0.002	21057	6475	1
alpha_late[46]	0.609	0.128	0.365	0.844	0.001	0.001	20913	6593	1
alpha_late[47]	-0.138	0.126	-0.372	0.101	0.001	0.001	21645	7323	1
alpha_late[48]	-0.468	0.128	-0.702	-0.221	0.001	0.002	22259	7283	1
alpha_late[49]	-0.384	0.125	-0.613	-0.148	0.001	0.001	22136	6879	1
alpha_late[50]	1.87	0.128	1.633	2.106	0.001	0.001	13465	7788	1
alpha_late[51]	0.799	0.127	0.566	1.03	0.001	0.001	18308	7547	1
alpha_late[52]	0.42	0.13	0.169	0.653	0.001	0.002	19571	6833	1
alpha_late[53]	0.552	0.127	0.318	0.795	0.001	0.001	19485	6881	1
alpha_late[54]	0.361	0.129	0.118	0.6	0.001	0.002	20721	6616	1
alpha_late[55]	-1.29	0.126	-1.531	-1.053	0.001	0.002	17143	6432	1
alpha_late[56]	-0.391	0.126	-0.627	-0.155	0.001	0.002	23373	6543	1
alpha_late[57]	-0.277	0.13	-0.532	-0.04	0.001	0.002	21842	6285	1
alpha_late[58]	-0.376	0.126	-0.614	-0.141	0.001	0.001	21291	7167	1
alpha_late[59]	1.248	0.126	1.02	1.492	0.001	0.001	15759	6596	1

alpha_late[60]	1.347	0.128	1.105	1.588	0.001	0.001	15797	7100	1
alpha_late[61]	2.1	0.126	1.874	2.344	0.001	0.001	11744	7736	1
alpha_late[62]	0.245	0.125	0.021	0.486	0.001	0.001	20365	6862	1
alpha_late[63]	-1.323	0.127	-1.552	-1.08	0.001	0.001	16120	7212	1
alpha_late[64]	-0.274	0.127	-0.514	-0.041	0.001	0.002	21599	7219	1
alpha_late[65]	0.267	0.128	0.021	0.501	0.001	0.001	20923	7269	1
alpha_late[66]	1.193	0.129	0.941	1.427	0.001	0.001	15740	7029	1
alpha_late[67]	-0.68	0.124	-0.912	-0.452	0.001	0.001	20884	7608	1
alpha_late[68]	-0.058	0.128	-0.293	0.186	0.001	0.001	20782	7164	1
alpha_late[69]	-0.03	0.127	-0.284	0.196	0.001	0.001	22601	6875	1
alpha_late[70]	-0.46	0.129	-0.7	-0.211	0.001	0.002	20903	6634	1
alpha_late[71]	-0.415	0.126	-0.644	-0.173	0.001	0.002	20027	6772	1
alpha_late[72]	1.721	0.129	1.488	1.965	0.001	0.002	12751	6967	1
alpha_late[73]	-0.421	0.13	-0.665	-0.177	0.001	0.002	21128	6393	1
alpha_late[74]	-0.545	0.129	-0.783	-0.3	0.001	0.002	21379	7220	1
alpha_late[75]	-1.186	0.127	-1.42	-0.944	0.001	0.002	18689	6907	1
alpha_late[76]	-1.168	0.124	-1.412	-0.949	0.001	0.002	15493	6350	1
alpha_late[77]	-1.183	0.127	-1.416	-0.945	0.001	0.001	17786	6407	1
alpha_late[78]	0.909	0.126	0.677	1.148	0.001	0.001	18719	7300	1
alpha_late[79]	0.146	0.125	-0.086	0.379	0.001	0.001	20600	7039	1
alpha_late[80]	-0.926	0.126	-1.161	-0.689	0.001	0.001	17434	7265	1
alpha_late[81]	-0.182	0.128	-0.433	0.047	0.001	0.002	22716	7671	1
alpha_late[82]	0.113	0.127	-0.12	0.358	0.001	0.002	22141	7396	1
alpha_late[83]	0.135	0.128	-0.097	0.382	0.001	0.001	22941	6975	1
alpha_late[84]	-0.781	0.129	-1.018	-0.536	0.001	0.002	21252	6031	1
alpha_late[85]	-1.144	0.128	-1.386	-0.907	0.001	0.001	18815	7606	1
alpha_late[86]	-1.779	0.131	-2.027	-1.539	0.001	0.001	13701	6977	1
alpha_late[87]	-0.372	0.127	-0.612	-0.135	0.001	0.002	20221	6921	1
alpha_late[88]	-0.537	0.128	-0.769	-0.29	0.001	0.001	21519	7092	1
alpha_late[89]	-1.246	0.128	-1.479	-0.995	0.001	0.001	18766	7228	1
alpha_late[90]	-0.978	0.127	-1.222	-0.749	0.001	0.001	17289	6902	1

alpha_late[91]	-1.359	0.129	-1.6	-1.105	0.001	0.002	16336	6436	1
alpha_late[92]	-0.813	0.127	-1.056	-0.581	0.001	0.001	19922	7296	1
alpha_late[93]	-0.986	0.128	-1.222	-0.737	0.001	0.002	18873	6355	1
alpha_late[94]	-1.207	0.124	-1.446	-0.98	0.001	0.001	19878	7544	1
alpha_late[95]	-0.18	0.126	-0.415	0.055	0.001	0.001	22181	6948	1
alpha_late[96]	0.096	0.125	-0.14	0.328	0.001	0.001	20937	6432	1
alpha_late[97]	-0.083	0.128	-0.32	0.157	0.001	0.001	20199	7308	1
alpha_late[98]	0.776	0.123	0.544	1.004	0.001	0.001	19946	7512	1
alpha_late[99]	-0.313	0.127	-0.557	-0.078	0.001	0.002	21261	6745	1
alpha_late[100]	-0.884	0.123	-1.113	-0.652	0.001	0.001	17962	7056	1
alpha_late[101]	-1.01	0.128	-1.25	-0.769	0.001	0.002	18705	7329	1
alpha_late[102]	-0.507	0.126	-0.745	-0.272	0.001	0.001	19809	7047	1
alpha_late[103]	-0.891	0.124	-1.123	-0.659	0.001	0.001	19771	7429	1
alpha_late[104]	0.202	0.128	-0.049	0.431	0.001	0.001	20587	6678	1
alpha_late[105]	-1.258	0.126	-1.485	-1.017	0.001	0.001	17765	7745	1
alpha_late[106]	-0.947	0.127	-1.182	-0.704	0.001	0.001	18326	7019	1
alpha_late[107]	0.591	0.125	0.363	0.838	0.001	0.001	20732	6826	1
alpha_late[108]	-0.074	0.125	-0.304	0.165	0.001	0.001	20264	7328	1
alpha_late[109]	-0.796	0.126	-1.027	-0.556	0.001	0.001	18090	6957	1
alpha_late[110]	-0.781	0.127	-1.022	-0.55	0.001	0.001	18787	7263	1
alpha_late[111]	-0.096	0.126	-0.337	0.134	0.001	0.002	19680	6707	1
alpha_late[112]	-0.131	0.127	-0.377	0.094	0.001	0.002	19853	6640	1
alpha_late[113]	-1.112	0.13	-1.351	-0.865	0.001	0.001	18547	6919	1
alpha_late[114]	-0.345	0.127	-0.58	-0.106	0.001	0.001	21403	7081	1
alpha_late[115]	-0.861	0.125	-1.095	-0.627	0.001	0.001	19323	6754	1
alpha_over[0]	0.912	0.228	0.489	1.339	0.001	0.003	26299	7150	1
alpha_over[1]	0.943	0.224	0.535	1.369	0.001	0.003	23410	7253	1
alpha_over[2]	0.474	0.231	0.047	0.901	0.001	0.003	25958	7328	1
alpha_over[3]	0.723	0.229	0.285	1.144	0.001	0.003	23401	6868	1
alpha_over[4]	0.832	0.231	0.411	1.281	0.002	0.003	22026	7887	1
alpha_over[5]	0.542	0.23	0.11	0.973	0.001	0.003	24913	6897	1

alpha_over[6]	0.601	0.227	0.187	1.055	0.001	0.003	27820	6564	1
alpha_over[7]	0.357	0.229	-0.057	0.796	0.001	0.003	23798	7690	1
alpha_over[8]	0.926	0.228	0.498	1.361	0.001	0.003	26573	7106	1
alpha_over[9]	0.659	0.231	0.219	1.087	0.001	0.003	25159	7285	1
alpha_over[10]	0.654	0.226	0.224	1.062	0.002	0.003	21789	6356	1
alpha_over[11]	0.587	0.222	0.172	1.006	0.001	0.003	24996	7216	1
alpha_over[12]	0.611	0.225	0.182	1.029	0.001	0.003	23618	6940	1
alpha_over[13]	0.45	0.223	0.029	0.866	0.001	0.003	26453	7052	1
alpha_over[14]	0.729	0.231	0.297	1.158	0.001	0.003	26353	7362	1
alpha_over[15]	-0.026	0.233	-0.469	0.403	0.002	0.003	21858	7075	1
alpha_over[16]	0.794	0.23	0.37	1.232	0.001	0.003	26983	6802	1
alpha_over[17]	0.081	0.231	-0.366	0.498	0.001	0.003	23872	7174	1
alpha_over[18]	0.333	0.23	-0.106	0.755	0.002	0.003	23389	6910	1
alpha_over[19]	0.549	0.227	0.135	0.986	0.002	0.003	22751	6421	1
alpha_over[20]	1.461	0.234	1.024	1.904	0.002	0.003	19904	7519	1
alpha_over[21]	0.473	0.226	0.031	0.883	0.001	0.003	26191	7618	1
alpha_over[22]	0.216	0.231	-0.224	0.637	0.001	0.003	26027	7157	1
alpha_over[23]	1.24	0.232	0.815	1.689	0.002	0.003	21335	7215	1
alpha_over[24]	0.281	0.231	-0.158	0.706	0.001	0.003	25234	7613	1
alpha_over[25]	0.291	0.231	-0.123	0.74	0.001	0.003	24770	7354	1
alpha_over[26]	0.755	0.231	0.322	1.187	0.001	0.003	24315	7413	1
alpha_over[27]	0.465	0.231	0.052	0.923	0.001	0.003	24221	7278	1
alpha_over[28]	0.847	0.229	0.427	1.28	0.002	0.003	21394	6414	1
alpha_over[29]	0.927	0.228	0.494	1.346	0.001	0.003	24567	7392	1
alpha_over[30]	0.667	0.227	0.218	1.073	0.002	0.003	21986	7108	1
alpha_over[31]	-0.312	0.232	-0.746	0.122	0.002	0.003	20700	7305	1
alpha_over[32]	1.083	0.23	0.65	1.516	0.002	0.003	23070	7025	1
alpha_over[33]	0.463	0.227	0.043	0.897	0.001	0.003	22993	6417	1
alpha_over[34]	0.138	0.227	-0.275	0.575	0.001	0.003	23466	7353	1
alpha_over[35]	0.484	0.233	0.044	0.922	0.001	0.003	25941	7294	1
alpha_over[36]	0.589	0.23	0.157	1.02	0.001	0.003	25165	6490	1

alpha_over[37]	0.464	0.229	0.039	0.9	0.001	0.003	24487	6474	1
alpha_over[38]	0.595	0.229	0.176	1.027	0.001	0.003	23904	6941	1
alpha_over[39]	0.161	0.231	-0.283	0.585	0.001	0.003	25851	7210	1
alpha_over[40]	0.85	0.226	0.43	1.28	0.002	0.003	22355	7215	1
alpha_over[41]	0.751	0.225	0.302	1.155	0.001	0.003	24438	6401	1
alpha_over[42]	0.578	0.227	0.151	0.999	0.001	0.002	23576	7498	1
alpha_over[43]	0.348	0.23	-0.073	0.782	0.002	0.003	22713	6254	1
alpha_over[44]	0.573	0.229	0.155	1.014	0.001	0.003	26459	7080	1
alpha_over[45]	0.581	0.227	0.169	1.003	0.001	0.003	23888	7511	1
alpha_over[46]	0.338	0.226	-0.089	0.751	0.001	0.003	25992	7214	1
alpha_over[47]	0.115	0.229	-0.299	0.554	0.002	0.002	21076	6717	1
alpha_over[48]	0.75	0.224	0.331	1.161	0.001	0.003	25690	7189	1
alpha_over[49]	1.149	0.23	0.722	1.57	0.001	0.002	23823	7580	1
alpha_over[50]	0.044	0.231	-0.387	0.488	0.001	0.003	23559	6914	1
alpha_over[51]	-0.239	0.235	-0.67	0.212	0.002	0.003	18962	6376	1
alpha_over[52]	0.541	0.226	0.118	0.977	0.002	0.003	21985	7236	1
alpha_over[53]	0.641	0.231	0.21	1.081	0.001	0.003	25974	7240	1
alpha_over[54]	-0.167	0.237	-0.605	0.275	0.002	0.003	20435	7145	1
alpha_over[55]	0.184	0.232	-0.229	0.643	0.002	0.003	23125	6767	1
alpha_over[56]	0.73	0.23	0.308	1.161	0.001	0.003	23758	7235	1
alpha_over[57]	0.567	0.23	0.154	1.014	0.002	0.003	23196	6737	1
alpha_over[58]	0.705	0.231	0.286	1.151	0.002	0.003	22911	7248	1
alpha_over[59]	0.704	0.226	0.271	1.111	0.001	0.002	22779	7200	1
alpha_over[60]	0.827	0.23	0.414	1.275	0.001	0.003	23742	7282	1
alpha_over[61]	0.843	0.229	0.421	1.279	0.002	0.003	22979	7434	1
alpha_over[62]	1.087	0.226	0.669	1.518	0.002	0.003	20403	6923	1
alpha_over[63]	-0.036	0.231	-0.477	0.379	0.002	0.003	22617	7084	1
alpha_over[64]	0.001	0.23	-0.433	0.429	0.001	0.003	26229	7440	1
alpha_over[65]	0.262	0.231	-0.157	0.7	0.001	0.003	24350	7024	1
alpha_over[66]	0.346	0.229	-0.085	0.773	0.001	0.003	23848	7140	1
alpha_over[67]	-0.309	0.226	-0.731	0.115	0.002	0.002	21191	8094	1

alpha_over[68]	-0.274	0.231	-0.719	0.157	0.002	0.003	19254	5997	1
alpha_over[69]	0.35	0.229	-0.067	0.793	0.001	0.003	24520	7183	1
alpha_over[70]	0.461	0.23	0.032	0.898	0.001	0.003	24061	7032	1
alpha_over[71]	-0.026	0.235	-0.449	0.429	0.002	0.003	20628	6766	1
alpha_over[72]	0.874	0.229	0.453	1.323	0.002	0.003	23119	6815	1
alpha_over[73]	1.112	0.229	0.68	1.537	0.002	0.003	22174	7197	1
alpha_over[74]	0.871	0.227	0.454	1.306	0.002	0.003	22529	7887	1
alpha_over[75]	0.345	0.227	-0.07	0.776	0.001	0.003	24731	7394	1
alpha_over[76]	0.903	0.233	0.465	1.336	0.002	0.003	22074	5971	1
alpha_over[77]	0.68	0.23	0.229	1.09	0.002	0.003	22674	7350	1
alpha_over[78]	0.31	0.227	-0.11	0.738	0.001	0.003	24890	6908	1
alpha_over[79]	0.772	0.227	0.325	1.181	0.002	0.003	21541	6725	1
alpha_over[80]	0.215	0.228	-0.212	0.633	0.001	0.003	23064	7059	1
alpha_over[81]	0.754	0.223	0.325	1.161	0.001	0.003	25289	7277	1
alpha_over[82]	0.47	0.224	0.042	0.882	0.001	0.003	24057	7575	1
alpha_over[83]	0.506	0.23	0.067	0.924	0.001	0.003	24310	7501	1
alpha_over[84]	1.239	0.226	0.83	1.677	0.002	0.003	18835	7471	1
alpha_over[85]	0.822	0.224	0.416	1.253	0.001	0.002	24256	7753	1
alpha_over[86]	0.748	0.227	0.32	1.165	0.001	0.003	24300	7137	1
alpha_over[87]	1.096	0.231	0.675	1.552	0.002	0.003	20677	5890	1
alpha_over[88]	1.08	0.231	0.647	1.518	0.002	0.003	23038	7151	1
alpha_over[89]	1.231	0.232	0.805	1.68	0.002	0.003	18554	7964	1
alpha_over[90]	0.371	0.228	-0.03	0.812	0.001	0.003	24756	7369	1
alpha_over[91]	1.024	0.229	0.584	1.445	0.002	0.003	22460	7468	1
alpha_over[92]	0.668	0.225	0.234	1.074	0.001	0.003	23992	7351	1
alpha_over[93]	1.058	0.229	0.627	1.483	0.002	0.003	22374	6968	1
alpha_over[94]	1.302	0.232	0.878	1.745	0.002	0.003	22233	7360	1
alpha_over[95]	0.253	0.23	-0.167	0.695	0.001	0.003	24922	7393	1
alpha_over[96]	0.47	0.226	0.031	0.879	0.001	0.003	24569	7419	1
alpha_over[97]	0.782	0.227	0.354	1.208	0.001	0.003	23857	7426	1
alpha_over[98]	0.671	0.224	0.263	1.101	0.002	0.003	21992	7127	1

alpha_over[99]	0.448	0.231	0.013	0.874	0.001	0.003	25729	7400	1
alpha_over[100]	0.917	0.226	0.488	1.332	0.001	0.003	24640	6979	1
alpha_over[101]	0.804	0.23	0.381	1.239	0.002	0.003	21436	6598	1
alpha_over[102]	0.452	0.221	0.041	0.871	0.001	0.002	22651	7549	1
alpha_over[103]	0.331	0.226	-0.092	0.749	0.001	0.003	24239	7039	1
alpha_over[104]	0.495	0.228	0.072	0.929	0.001	0.003	26450	6915	1
alpha_over[105]	0.385	0.229	-0.05	0.818	0.001	0.003	25114	6838	1
alpha_over[106]	0.405	0.229	-0.034	0.825	0.001	0.003	23552	7544	1
alpha_over[107]	-0.061	0.229	-0.485	0.376	0.001	0.003	23665	6664	1
alpha_over[108]	0.385	0.227	-0.032	0.824	0.002	0.003	22676	7418	1
alpha_over[109]	1.022	0.228	0.597	1.446	0.002	0.003	22043	6808	1
alpha_over[110]	0.225	0.229	-0.221	0.644	0.001	0.003	25179	7032	1
alpha_over[111]	0.579	0.23	0.158	1.016	0.001	0.003	25784	6804	1
alpha_over[112]	0.646	0.225	0.232	1.072	0.002	0.003	21855	7262	1
alpha_over[113]	0.656	0.228	0.231	1.082	0.001	0.003	24438	6610	1
alpha_over[114]	0.438	0.224	0.01	0.846	0.001	0.003	26661	6925	1
alpha_over[115]	0.664	0.228	0.233	1.084	0.002	0.003	21576	6787	1
b_late_over	0.037	0.22	-0.376	0.445	0.003	0.002	4957	7040	1
beta_late[0]	-0.032	0.049	-0.124	0.062	0	0.001	16906	7978	1
beta_late[1]	0.164	0.048	0.072	0.253	0	0	17864	8040	1
beta_late[2]	0.208	0.048	0.115	0.295	0	0	17848	7486	1
beta_late[3]	0.011	0.047	-0.076	0.1	0	0.001	18561	8125	1
beta_late[4]	0.017	0.047	-0.073	0.103	0	0.001	22446	7405	1
beta_late[5]	0.127	0.048	0.039	0.218	0	0.001	21368	7165	1
beta_late[6]	0.179	0.048	0.085	0.266	0	0.001	18190	7925	1
beta_late[7]	0.068	0.048	-0.025	0.156	0	0.001	18717	7290	1
beta_late[8]	0.191	0.048	0.103	0.281	0	0.001	18120	7251	1
beta_late[9]	0.05	0.048	-0.042	0.139	0	0.001	18275	7232	1
beta_late[10]	0.11	0.047	0.018	0.193	0	0.001	22858	7836	1
beta_late[11]	0.135	0.048	0.045	0.224	0	0	17733	8234	1
beta_late[12]	0.004	0.049	-0.084	0.097	0	0.001	16445	7316	1

beta_late[13]	0.199	0.048	0.109	0.288	0	0.001	18839	8026	1
beta_late[14]	0.135	0.048	0.043	0.223	0	0.001	20450	7065	1
beta_late[15]	0.069	0.048	-0.023	0.158	0	0	19875	7924	1
beta_late[16]	0.169	0.049	0.078	0.26	0	0	15888	7959	1
beta_late[17]	0.244	0.051	0.147	0.339	0	0	13785	9056	1
beta_late[18]	0.118	0.047	0.031	0.208	0	0.001	17384	7351	1
beta_late[19]	0.12	0.05	0.027	0.212	0	0	14629	8649	1
beta_late[20]	0.14	0.049	0.052	0.235	0	0.001	20359	6708	1
beta_late[21]	0.409	0.049	0.315	0.5	0	0	13256	7555	1
beta_late[22]	0.049	0.048	-0.042	0.136	0	0.001	17785	6965	1
beta_late[23]	0.002	0.048	-0.088	0.092	0	0.001	19193	7421	1
beta_late[24]	0.083	0.048	-0.004	0.178	0	0.001	23107	7546	1
beta_late[25]	0.058	0.047	-0.029	0.145	0	0.001	20117	7821	1
beta_late[26]	0.17	0.049	0.077	0.258	0	0.001	17582	6542	1
beta_late[27]	-0.004	0.048	-0.093	0.086	0	0.001	17855	7696	1
beta_late[28]	0.047	0.049	-0.043	0.14	0	0	14240	8294	1
beta_late[29]	0.072	0.048	-0.02	0.161	0	0.001	19191	7733	1
beta_late[30]	0.172	0.048	0.086	0.264	0	0	18370	8408	1
beta_late[31]	0.144	0.048	0.058	0.237	0	0	17082	8490	1
beta_late[32]	0.062	0.048	-0.029	0.153	0	0.001	22315	7497	1
beta_late[33]	0.12	0.049	0.026	0.213	0	0	16566	7936	1
beta_late[34]	0.165	0.049	0.071	0.254	0	0.001	20967	7113	1
beta_late[35]	0.204	0.048	0.111	0.294	0	0.001	21644	7055	1
beta_late[36]	0.169	0.048	0.077	0.256	0	0.001	17505	8446	1
beta_late[37]	0.139	0.047	0.05	0.229	0	0.001	23063	7830	1
beta_late[38]	0.039	0.048	-0.051	0.129	0	0	18471	8308	1
beta_late[39]	0.046	0.047	-0.048	0.131	0	0.001	20250	7498	1
beta_late[40]	0.157	0.048	0.068	0.245	0	0	20760	7766	1
beta_late[41]	0.125	0.048	0.037	0.217	0	0.001	21516	8102	1
beta_late[42]	-0.001	0.048	-0.09	0.09	0	0.001	19717	6819	1
beta_late[43]	0.156	0.047	0.071	0.247	0	0.001	18216	6944	1

beta_late[44]	0.161	0.048	0.074	0.253	0	0.001	18575	7779	1
beta_late[45]	0.046	0.048	-0.044	0.135	0	0.001	19753	6466	1
beta_late[46]	0.099	0.049	0.01	0.191	0	0	17910	7894	1
beta_late[47]	0.116	0.047	0.029	0.204	0	0.001	20568	7386	1
beta_late[48]	0.19	0.05	0.096	0.283	0	0	13618	8462	1
beta_late[49]	0.118	0.047	0.024	0.202	0	0.001	20732	7543	1
beta_late[50]	0.02	0.048	-0.067	0.113	0	0.001	21232	7633	1
beta_late[51]	0.175	0.048	0.087	0.269	0	0.001	19138	6700	1
beta_late[52]	0.063	0.048	-0.026	0.152	0	0.001	21233	6477	1
beta_late[53]	0.07	0.048	-0.019	0.159	0	0.001	20448	7079	1
beta_late[54]	0.116	0.047	0.028	0.207	0	0.001	21882	6980	1
beta_late[55]	0.009	0.048	-0.078	0.102	0	0.001	19635	7027	1
beta_late[56]	0.065	0.046	-0.021	0.152	0	0	22187	7982	1
beta_late[57]	0.039	0.048	-0.051	0.128	0	0.001	20011	7404	1
beta_late[58]	0.212	0.047	0.124	0.3	0	0.001	19913	7333	1
beta_late[59]	0.171	0.047	0.08	0.256	0	0.001	20036	6689	1
beta_late[60]	0.136	0.048	0.044	0.224	0	0.001	21166	6384	1
beta_late[61]	0.162	0.049	0.07	0.251	0	0.001	21497	7432	1
beta_late[62]	0.093	0.048	0.004	0.182	0	0.001	21937	6625	1
beta_late[63]	0.158	0.048	0.065	0.245	0	0.001	19710	6829	1
beta_late[64]	0.112	0.05	0.021	0.21	0	0	13497	7149	1
beta_late[65]	0.398	0.05	0.305	0.491	0	0	13091	8417	1
beta_late[66]	-0.127	0.049	-0.224	-0.036	0	0	14221	8314	1
beta_late[67]	0.047	0.049	-0.044	0.137	0	0.001	19910	7219	1
beta_late[68]	0.182	0.05	0.089	0.278	0	0	14514	8646	1
beta_late[69]	0.22	0.05	0.126	0.313	0	0	13724	8830	1
beta_late[70]	-0.015	0.047	-0.106	0.072	0	0.001	18940	7340	1
beta_late[71]	0.154	0.049	0.063	0.243	0	0.001	21290	7425	1
beta_late[72]	0.119	0.047	0.03	0.207	0	0.001	23258	7028	1
beta_late[73]	0.136	0.048	0.042	0.219	0	0.001	20454	7280	1
beta_late[74]	0.131	0.05	0.041	0.227	0	0	14759	8093	1

beta_late[75]	0.079	0.047	-0.01	0.17	0	0.001	19705	7532	1
beta_late[76]	0.088	0.047	0.003	0.177	0	0.001	21102	7601	1
beta_late[77]	0.096	0.048	0.006	0.183	0	0	16612	8099	1
beta_late[78]	0.13	0.049	0.039	0.224	0	0	14982	7159	1
beta_late[79]	-0.023	0.047	-0.115	0.063	0	0.001	19182	8018	1
beta_late[80]	-0.115	0.049	-0.208	-0.025	0	0.001	18886	7367	1
beta_late[81]	0.115	0.048	0.024	0.203	0	0.001	23444	6671	1
beta_late[82]	-0.067	0.048	-0.156	0.025	0	0.001	23239	8035	1
beta_late[83]	0.096	0.048	0.005	0.186	0	0.001	21543	6715	1
beta_late[84]	0.099	0.049	0.008	0.194	0	0	14580	8483	1
beta_late[85]	0.086	0.048	-0.004	0.175	0	0	19563	8533	1
beta_late[86]	0.134	0.047	0.044	0.22	0	0.001	21243	7291	1
beta_late[87]	0.068	0.049	-0.023	0.161	0	0	18750	8160	1
beta_late[88]	0.114	0.048	0.025	0.205	0	0.001	21937	6942	1
beta_late[89]	-0.05	0.048	-0.143	0.039	0	0	18927	7532	1
beta_late[90]	0.093	0.048	-0.001	0.182	0	0	18780	7736	1
beta_late[91]	0.017	0.048	-0.072	0.11	0	0.001	20254	7389	1
beta_late[92]	-0.029	0.049	-0.121	0.061	0	0	17629	8056	1
beta_late[93]	0.096	0.048	0.007	0.188	0	0.001	22739	7099	1
beta_late[94]	-0.052	0.05	-0.143	0.044	0	0	15911	9177	1
beta_late[95]	-0.119	0.05	-0.215	-0.026	0	0	13077	8778	1
beta_late[96]	0.187	0.05	0.093	0.28	0	0	15381	8512	1
beta_late[97]	0.206	0.049	0.116	0.298	0	0.001	19766	7132	1
beta_late[98]	0.144	0.048	0.051	0.23	0	0	19328	8197	1
beta_late[99]	0.031	0.047	-0.055	0.122	0	0.001	20489	7405	1
beta_late[100]	0.098	0.047	0.007	0.183	0	0	18843	7890	1
beta_late[101]	0.103	0.047	0.015	0.19	0	0.001	20476	7709	1
beta_late[102]	0.084	0.048	-0.006	0.172	0	0.001	21289	7673	1
beta_late[103]	0.133	0.047	0.049	0.226	0	0.001	19673	6810	1
beta_late[104]	0.068	0.048	-0.024	0.156	0	0	18771	8108	1
beta_late[105]	0.09	0.047	-0.001	0.177	0	0.001	21142	8070	1

beta_late[106]	0.056	0.049	-0.032	0.15	0	0.001	19576	7502	1
beta_late[107]	0.13	0.048	0.041	0.22	0	0.001	18854	7256	1
beta_late[108]	0.093	0.05	-0.004	0.183	0	0	14402	8275	1
beta_late[109]	0.069	0.049	-0.022	0.162	0	0.001	21253	8127	1
beta_late[110]	0.073	0.048	-0.017	0.164	0	0.001	20288	7067	1
beta_late[111]	0.139	0.047	0.048	0.225	0	0.001	20134	7177	1
beta_late[112]	0.037	0.048	-0.058	0.122	0	0.001	20478	7380	1
beta_late[113]	0.105	0.05	0.008	0.196	0	0.001	19665	7783	1
beta_late[114]	0.087	0.049	-0.002	0.18	0	0.001	22899	7492	1
beta_late[115]	0.112	0.047	0.027	0.202	0	0.001	21714	7368	1
beta_late_base[0]	0.01	0.053	-0.094	0.105	0	0.001	13168	8154	1
beta_late_base[1]	0.161	0.05	0.064	0.252	0	0	15737	7895	1
beta_late_base[2]	0.186	0.056	0.075	0.286	0.001	0.001	10094	7642	1
beta_late_base[3]	0.008	0.047	-0.08	0.098	0	0.001	17951	8516	1
beta_late_base[4]	0.038	0.049	-0.055	0.126	0	0.001	18562	7091	1
beta_late_base[5]	0.149	0.054	0.046	0.249	0.001	0.001	10692	7216	1
beta_late_base[6]	0.191	0.056	0.083	0.292	0.001	0.001	9982	7918	1
beta_late_base[7]	0.063	0.054	-0.04	0.161	0.001	0.001	11369	7720	1
beta_late_base[8]	0.191	0.054	0.095	0.294	0	0.001	12071	7588	1
beta_late_base[9]	0.07	0.056	-0.03	0.177	0.001	0.001	11167	7548	1
beta_late_base[10]	0.101	0.048	0.008	0.187	0	0.001	22077	7683	1
beta_late_base[11]	0.082	0.058	-0.026	0.188	0.001	0.001	9965	8147	1
beta_late_base[12]	-0.02	0.048	-0.11	0.07	0	0.001	18049	7557	1
beta_late_base[13]	0.107	0.055	-0.002	0.205	0.001	0.001	11657	8027	1
beta_late_base[14]	0.109	0.051	0.016	0.209	0	0.001	14799	7414	1
beta_late_base[15]	0.013	0.053	-0.088	0.109	0	0	13369	8012	1
beta_late_base[16]	0.042	0.061	-0.068	0.162	0.001	0.001	9038	7113	1
beta_late_base[17]	0.108	0.066	-0.017	0.23	0.001	0.001	10824	8223	1
beta_late_base[18]	0.073	0.055	-0.03	0.174	0.001	0.001	10266	6577	1
beta_late_base[19]	0.048	0.061	-0.067	0.16	0.001	0.001	9134	7877	1
beta_late_base[20]	0.093	0.052	-0.006	0.188	0	0.001	16159	7026	1

beta_late_base[21]	0.28	0.06	0.171	0.397	0.001	0.001	9453	8168	1
beta_late_base[22]	0.008	0.053	-0.093	0.105	0	0.001	12698	7598	1
beta_late_base[23]	0.048	0.056	-0.055	0.151	0.001	0.001	10235	8136	1
beta_late_base[24]	0.098	0.049	0.006	0.19	0	0.001	20865	7526	1
beta_late_base[25]	0.078	0.049	-0.014	0.17	0	0.001	15683	8337	1
beta_late_base[26]	0.213	0.052	0.116	0.311	0	0.001	12851	7780	1
beta_late_base[27]	0.011	0.054	-0.088	0.112	0.001	0.001	10360	7750	1
beta_late_base[28]	0.064	0.064	-0.052	0.188	0.001	0.001	8410	8092	1
beta_late_base[29]	0.099	0.058	-0.012	0.205	0.001	0.001	9665	7244	1
beta_late_base[30]	0.151	0.055	0.048	0.253	0.001	0	10111	7423	1
beta_late_base[31]	0.077	0.055	-0.027	0.18	0.001	0.001	10462	7857	1
beta_late_base[32]	0.057	0.049	-0.035	0.15	0	0.001	20023	7297	1
beta_late_base[33]	0.077	0.06	-0.035	0.192	0.001	0.001	8860	8146	1
beta_late_base[34]	0.164	0.051	0.07	0.26	0	0.001	18415	7255	1
beta_late_base[35]	0.167	0.05	0.075	0.264	0	0.001	19169	7576	1
beta_late_base[36]	0.121	0.056	0.012	0.22	0.001	0.001	10541	8146	1
beta_late_base[37]	0.11	0.049	0.017	0.203	0	0.001	19025	8161	1
beta_late_base[38]	0.071	0.058	-0.041	0.177	0.001	0.001	8888	8020	1
beta_late_base[39]	0.057	0.05	-0.036	0.152	0	0.001	15691	6856	1
beta_late_base[40]	0.166	0.056	0.061	0.271	0.001	0.001	10116	8147	1
beta_late_base[41]	0.119	0.049	0.027	0.21	0	0.001	19741	7651	1
beta_late_base[42]	0.012	0.054	-0.091	0.113	0.001	0.001	10822	7616	1
beta_late_base[43]	0.132	0.051	0.035	0.227	0	0	13330	8290	1
beta_late_base[44]	0.193	0.054	0.09	0.294	0.001	0.001	10417	8145	1
beta_late_base[45]	0.044	0.053	-0.057	0.144	0	0.001	12447	7827	1
beta_late_base[46]	0.086	0.057	-0.019	0.194	0.001	0.001	10271	8016	1
beta_late_base[47]	0.099	0.052	0.004	0.2	0	0.001	11984	7641	1
beta_late_base[48]	0.159	0.066	0.038	0.286	0.001	0.001	9375	8158	1
beta_late_base[49]	0.11	0.054	0.005	0.208	0	0.001	11587	7759	1
beta_late_base[50]	0.037	0.053	-0.062	0.139	0.001	0.001	11147	7320	1
beta_late_base[51]	0.184	0.052	0.088	0.286	0	0.001	12509	7094	1

beta_late_base[52]	0.078	0.054	-0.025	0.179	0.001	0.001	10775	7691	1
beta_late_base[53]	0.072	0.054	-0.036	0.17	0.001	0.001	11122	7201	1
beta_late_base[54]	0.11	0.048	0.024	0.204	0	0.001	21059	6939	1
beta_late_base[55]	0.005	0.055	-0.096	0.11	0.001	0.001	10067	7088	1
beta_late_base[56]	0.086	0.054	-0.015	0.186	0.001	0	10212	7869	1
beta_late_base[57]	0.035	0.054	-0.069	0.136	0.001	0.001	10487	7514	1
beta_late_base[58]	0.183	0.047	0.092	0.269	0	0.001	20392	7246	1
beta_late_base[59]	0.199	0.053	0.102	0.301	0	0.001	11034	7364	1
beta_late_base[60]	0.128	0.054	0.031	0.234	0.001	0.001	11518	7641	1
beta_late_base[61]	0.2	0.055	0.097	0.302	0.001	0.001	11053	7957	1
beta_late_base[62]	0.081	0.053	-0.023	0.179	0	0.001	12766	7378	1
beta_late_base[63]	0.15	0.053	0.052	0.252	0	0.001	12454	7128	1
beta_late_base[64]	0.087	0.061	-0.027	0.203	0.001	0.001	9583	8449	1
beta_late_base[65]	0.382	0.054	0.282	0.485	0.001	0	10295	8676	1
beta_late_base[66]	-0.091	0.054	-0.195	0.009	0.001	0	9728	8501	1
beta_late_base[67]	0.093	0.055	-0.013	0.196	0.001	0.001	10448	7134	1
beta_late_base[68]	0.085	0.065	-0.031	0.208	0.001	0.001	9933	8541	1
beta_late_base[69]	0.112	0.065	-0.01	0.234	0.001	0.001	10121	8491	1
beta_late_base[70]	0.005	0.054	-0.097	0.106	0.001	0.001	10213	7290	1
beta_late_base[71]	0.127	0.055	0.021	0.23	0.001	0.001	10717	7967	1
beta_late_base[72]	0.128	0.053	0.028	0.227	0	0.001	12331	7715	1
beta_late_base[73]	0.126	0.051	0.025	0.217	0	0.001	14793	8211	1
beta_late_base[74]	0.124	0.064	0.002	0.24	0.001	0.001	10076	7714	1
beta_late_base[75]	0.099	0.048	0.005	0.186	0	0.001	18916	7991	1
beta_late_base[76]	0.107	0.051	0.013	0.201	0	0.001	15132	8276	1
beta_late_base[77]	0.228	0.053	0.126	0.327	0.001	0	10707	8200	1
beta_late_base[78]	0.24	0.058	0.13	0.347	0.001	0.001	11445	8787	1
beta_late_base[79]	0.044	0.049	-0.048	0.135	0	0.001	15486	7786	1
beta_late_base[80]	-0.028	0.054	-0.13	0.07	0.001	0.001	11306	7808	1
beta_late_base[81]	0.17	0.053	0.065	0.265	0	0.001	11797	7256	1
beta_late_base[82]	0.034	0.055	-0.067	0.138	0.001	0.001	10954	7703	1

beta_late_base[83]	0.134	0.052	0.039	0.236	0	0.001	15793	8165	1
beta_late_base[84]	0.105	0.062	-0.013	0.218	0.001	0.001	9697	8075	1
beta_late_base[85]	0.123	0.057	0.017	0.228	0.001	0.001	9471	7746	1
beta_late_base[86]	0.193	0.053	0.09	0.288	0	0.001	11817	7633	1
beta_late_base[87]	0.118	0.055	0.017	0.223	0	0.001	14254	8282	1
beta_late_base[88]	0.148	0.053	0.051	0.249	0	0.001	15401	7868	1
beta_late_base[89]	0.085	0.057	-0.017	0.195	0.001	0.001	9071	7899	1
beta_late_base[90]	0.056	0.053	-0.042	0.156	0	0.001	12905	7461	1
beta_late_base[91]	0.063	0.054	-0.041	0.161	0	0.001	12340	7412	1
beta_late_base[92]	0.03	0.054	-0.075	0.128	0	0.001	13471	8472	1
beta_late_base[93]	0.131	0.051	0.028	0.224	0	0.001	14212	7699	1
beta_late_base[94]	0.061	0.063	-0.056	0.183	0.001	0.001	10936	8440	1
beta_late_base[95]	0.068	0.065	-0.051	0.194	0.001	0.001	10630	8909	1
beta_late_base[96]	0.151	0.064	0.031	0.272	0.001	0.001	9030	8067	1
beta_late_base[97]	0.129	0.051	0.031	0.221	0	0.001	15410	6732	1
beta_late_base[98]	0.159	0.055	0.054	0.26	0.001	0	9964	8162	1
beta_late_base[99]	0.071	0.05	-0.018	0.168	0	0.001	15150	7617	1
beta_late_base[100]	0.102	0.056	-0.003	0.206	0.001	0	9729	7706	1
beta_late_base[101]	0.114	0.05	0.02	0.204	0	0.001	16675	8311	1
beta_late_base[102]	0.157	0.051	0.062	0.253	0	0.001	13584	8106	1
beta_late_base[103]	0.167	0.049	0.075	0.259	0	0.001	15753	6453	1
beta_late_base[104]	0.075	0.055	-0.025	0.181	0.001	0	10372	8537	1
beta_late_base[105]	0.05	0.049	-0.04	0.145	0	0.001	17420	7872	1
beta_late_base[106]	0.042	0.056	-0.063	0.144	0.001	0.001	11080	8059	1
beta_late_base[107]	0.101	0.056	-0.004	0.207	0.001	0.001	11105	8074	1
beta_late_base[108]	0.066	0.064	-0.053	0.183	0.001	0.001	9126	8840	1
beta_late_base[109]	0.08	0.06	-0.036	0.189	0.001	0.001	9293	7382	1
beta_late_base[110]	0.043	0.05	-0.049	0.138	0	0.001	17774	7490	1
beta_late_base[111]	0.122	0.049	0.029	0.212	0	0.001	18063	7523	1
beta_late_base[112]	0.133	0.053	0.031	0.229	0.001	0	11257	7780	1
beta_late_base[113]	0.141	0.058	0.025	0.244	0.001	0.001	9978	7441	1

beta_late_base[114]	0.084	0.05	-0.011	0.178	0	0.001	19241	7917	1
beta_late_base[115]	0.09	0.048	0	0.18	0	0.001	19048	7450	1
beta_over[0]	0.345	0.091	0.172	0.509	0.001	0.001	15290	8662	1
beta_over[1]	0.301	0.09	0.124	0.46	0.001	0.001	18579	7758	1
beta_over[2]	0.141	0.089	-0.033	0.3	0.001	0.001	16869	8351	1
beta_over[3]	0.17	0.086	0.012	0.332	0.001	0.001	22167	7239	1
beta_over[4]	0.039	0.087	-0.125	0.203	0.001	0.001	18851	7713	1
beta_over[5]	0.127	0.086	-0.028	0.292	0.001	0.001	25067	7890	1
beta_over[6]	0.123	0.088	-0.038	0.293	0.001	0.001	20876	6418	1
beta_over[7]	-0.003	0.091	-0.178	0.163	0.001	0.001	15829	7422	1
beta_over[8]	0.168	0.089	0.004	0.34	0.001	0.001	19288	7720	1
beta_over[9]	0.124	0.089	-0.037	0.299	0.001	0.001	18953	8025	1
beta_over[10]	0.074	0.085	-0.092	0.23	0.001	0.001	22680	7621	1
beta_over[11]	0.088	0.089	-0.085	0.249	0.001	0.001	18853	7738	1
beta_over[12]	0.166	0.086	0.001	0.321	0.001	0.001	19183	7681	1
beta_over[13]	0.031	0.089	-0.131	0.206	0.001	0.001	16904	8156	1
beta_over[14]	-0.138	0.09	-0.313	0.024	0.001	0.001	16694	7892	1
beta_over[15]	0.017	0.089	-0.144	0.189	0.001	0.001	16415	7390	1
beta_over[16]	0.128	0.093	-0.048	0.302	0.001	0.001	16360	8419	1
beta_over[17]	-0.056	0.098	-0.239	0.13	0.001	0.001	14688	8331	1
beta_over[18]	0.162	0.089	-0.007	0.328	0.001	0.001	19820	8366	1
beta_over[19]	0.136	0.092	-0.036	0.307	0.001	0.001	15226	8437	1
beta_over[20]	0.099	0.086	-0.062	0.258	0.001	0.001	19960	7580	1
beta_over[21]	0.149	0.094	-0.026	0.329	0.001	0.001	14812	8760	1
beta_over[22]	0.148	0.086	-0.01	0.315	0.001	0.001	20119	7307	1
beta_over[23]	0.051	0.089	-0.111	0.225	0.001	0.001	18704	7771	1
beta_over[24]	0.008	0.086	-0.156	0.164	0.001	0.001	21986	7155	1
beta_over[25]	-0.19	0.092	-0.368	-0.023	0.001	0.001	17027	7687	1
beta_over[26]	-0.126	0.089	-0.293	0.04	0.001	0.001	17137	7684	1
beta_over[27]	-0.093	0.09	-0.26	0.075	0.001	0.001	17527	7266	1
beta_over[28]	-0.146	0.095	-0.323	0.033	0.001	0.001	13260	8617	1

beta_over[29]	0.112	0.088	-0.057	0.269	0.001	0.001	20047	7892	1
beta_over[30]	0.058	0.089	-0.107	0.226	0.001	0.001	20119	7865	1
beta_over[31]	0.151	0.088	-0.012	0.314	0.001	0.001	19250	8162	1
beta_over[32]	0.196	0.087	0.035	0.363	0.001	0.001	20727	7683	1
beta_over[33]	0.097	0.092	-0.073	0.271	0.001	0.001	18081	8399	1
beta_over[34]	0.165	0.085	0.004	0.324	0.001	0.001	18118	7922	1
beta_over[35]	-0.02	0.087	-0.185	0.142	0.001	0.001	18430	7884	1
beta_over[36]	0.088	0.089	-0.081	0.254	0.001	0.001	17658	7820	1
beta_over[37]	0.119	0.086	-0.042	0.281	0.001	0.001	20789	6712	1
beta_over[38]	0.047	0.089	-0.12	0.213	0.001	0.001	17947	7747	1
beta_over[39]	-0.002	0.084	-0.155	0.161	0.001	0.001	20621	7356	1
beta_over[40]	0.141	0.089	-0.029	0.308	0.001	0.001	19075	7658	1
beta_over[41]	0.196	0.088	0.027	0.354	0.001	0.001	22348	6727	1
beta_over[42]	0.073	0.086	-0.084	0.239	0.001	0.001	20053	6681	1
beta_over[43]	0.305	0.088	0.14	0.466	0.001	0.001	18057	6829	1
beta_over[44]	0.253	0.088	0.092	0.42	0.001	0.001	22477	7432	1
beta_over[45]	0.251	0.086	0.098	0.419	0.001	0.001	18103	6738	1
beta_over[46]	0.168	0.089	0.005	0.338	0.001	0.001	19326	8359	1
beta_over[47]	0.236	0.087	0.069	0.394	0.001	0.001	20306	7101	1
beta_over[48]	0.556	0.098	0.373	0.746	0.001	0.001	14043	8428	1
beta_over[49]	0.256	0.089	0.089	0.422	0.001	0.001	18438	7654	1
beta_over[50]	0.194	0.087	0.024	0.348	0.001	0.001	22877	7580	1
beta_over[51]	0.063	0.087	-0.099	0.225	0.001	0.001	16745	7564	1
beta_over[52]	0.082	0.085	-0.076	0.24	0.001	0.001	21050	7658	1
beta_over[53]	0.175	0.086	0.017	0.339	0.001	0.001	20725	7035	1
beta_over[54]	0.071	0.085	-0.092	0.225	0.001	0.001	21496	7461	1
beta_over[55]	0.175	0.086	0.005	0.331	0.001	0.001	21210	7593	1
beta_over[56]	0.168	0.086	0.006	0.328	0.001	0.001	20808	7926	1
beta_over[57]	0.142	0.086	-0.018	0.304	0.001	0.001	22137	7456	1
beta_over[58]	0.189	0.088	0.027	0.352	0.001	0.001	20158	7713	1
beta_over[59]	0.123	0.085	-0.036	0.283	0.001	0.001	20789	7652	1

beta_over[60]	0.193	0.088	0.031	0.362	0.001	0.001	21495	7363	1
beta_over[61]	0.246	0.086	0.086	0.41	0.001	0.001	21057	8136	1
beta_over[62]	0.195	0.085	0.035	0.353	0.001	0.001	22848	7585	1
beta_over[63]	0.063	0.084	-0.093	0.219	0.001	0.001	21357	7848	1
beta_over[64]	0.12	0.093	-0.05	0.3	0.001	0.001	14146	9035	1
beta_over[65]	0.063	0.094	-0.114	0.237	0.001	0.001	14314	8628	1
beta_over[66]	0.119	0.091	-0.058	0.283	0.001	0.001	16233	7707	1
beta_over[67]	0.069	0.087	-0.098	0.23	0.001	0.001	20200	6908	1
beta_over[68]	0.151	0.096	-0.033	0.328	0.001	0.001	14712	8988	1
beta_over[69]	0.209	0.096	0.029	0.386	0.001	0.001	13596	8608	1
beta_over[70]	0.149	0.087	-0.009	0.315	0.001	0.001	21365	7369	1
beta_over[71]	0.167	0.088	0.012	0.339	0.001	0.001	21361	7935	1
beta_over[72]	0.09	0.085	-0.073	0.251	0.001	0.001	22684	7893	1
beta_over[73]	0.341	0.087	0.181	0.504	0.001	0.001	18413	8479	1
beta_over[74]	0.344	0.094	0.169	0.52	0.001	0.001	13914	7763	1
beta_over[75]	0.11	0.086	-0.049	0.276	0.001	0.001	20596	7144	1
beta_over[76]	0.132	0.088	-0.04	0.289	0.001	0.001	20510	7881	1
beta_over[77]	0.115	0.09	-0.054	0.284	0.001	0.001	17665	7821	1
beta_over[78]	-0.041	0.092	-0.207	0.14	0.001	0.001	14993	8681	1
beta_over[79]	0.125	0.089	-0.036	0.296	0.001	0.001	21283	8175	1
beta_over[80]	0.15	0.087	-0.012	0.312	0.001	0.001	19814	7518	1
beta_over[81]	0.162	0.087	-0.001	0.326	0.001	0.001	22081	7433	1
beta_over[82]	0.073	0.085	-0.081	0.238	0.001	0.001	19105	8211	1
beta_over[83]	0.248	0.088	0.081	0.414	0.001	0.001	21462	7759	1
beta_over[84]	0.258	0.095	0.084	0.441	0.001	0.001	14700	8312	1
beta_over[85]	0.111	0.089	-0.05	0.282	0.001	0.001	19245	7235	1
beta_over[86]	0.255	0.088	0.095	0.429	0.001	0.001	20349	8039	1
beta_over[87]	0.228	0.092	0.058	0.402	0.001	0.001	18145	7858	1
beta_over[88]	0.181	0.085	0.022	0.341	0.001	0.001	21545	7637	1
beta_over[89]	0.175	0.088	0.008	0.339	0.001	0.001	17061	7864	1
beta_over[90]	0.24	0.089	0.076	0.409	0.001	0.001	18735	8306	1

beta_over[91]	0.142	0.085	-0.013	0.306	0.001	0.001	21294	7150	1
beta_over[92]	0.093	0.088	-0.074	0.256	0.001	0.001	19323	7231	1
beta_over[93]	0.214	0.087	0.052	0.381	0.001	0.001	20375	7325	1
beta_over[94]	0.446	0.094	0.276	0.627	0.001	0.001	13783	9002	1
beta_over[95]	0.221	0.097	0.04	0.402	0.001	0.001	13850	8949	1
beta_over[96]	0.1	0.094	-0.076	0.277	0.001	0.001	15662	8959	1
beta_over[97]	0.069	0.088	-0.096	0.231	0.001	0.001	20548	7832	1
beta_over[98]	0.152	0.089	-0.005	0.328	0.001	0.001	18458	7758	1
beta_over[99]	0.043	0.087	-0.117	0.213	0.001	0.001	22729	7173	1
beta_over[100]	0.07	0.087	-0.09	0.233	0.001	0.001	19104	7582	1
beta_over[101]	0.025	0.086	-0.138	0.186	0.001	0.001	21386	7827	1
beta_over[102]	0.02	0.087	-0.141	0.185	0.001	0.001	23026	8015	1
beta_over[103]	0.072	0.084	-0.093	0.225	0.001	0.001	19688	7851	1
beta_over[104]	0.12	0.088	-0.046	0.282	0.001	0.001	17690	8342	1
beta_over[105]	0.016	0.086	-0.14	0.182	0.001	0.001	19697	6822	1
beta_over[106]	0.038	0.089	-0.126	0.209	0.001	0.001	19284	7865	1
beta_over[107]	0.023	0.088	-0.145	0.185	0.001	0.001	20971	8622	1
beta_over[108]	0.133	0.095	-0.043	0.308	0.001	0.001	16165	8607	1
beta_over[109]	0.05	0.09	-0.116	0.218	0.001	0.001	20757	8365	1
beta_over[110]	0.122	0.087	-0.045	0.282	0.001	0.001	19192	7536	1
beta_over[111]	0.092	0.084	-0.065	0.254	0.001	0.001	21910	7797	1
beta_over[112]	0.051	0.088	-0.119	0.212	0.001	0.001	22743	7660	1
beta_over[113]	0.097	0.087	-0.055	0.275	0.001	0.001	17980	7851	1
beta_over[114]	0.092	0.086	-0.071	0.251	0.001	0.001	19583	7554	1
beta_over[115]	0.038	0.086	-0.123	0.202	0.001	0.001	24213	7512	1
beta_over_base[0]	0.234	0.091	0.061	0.403	0.001	0.001	13550	8594	1
beta_over_base[1]	0.174	0.092	0.003	0.346	0.001	0.001	13308	7439	1
beta_over_base[2]	0.075	0.099	-0.117	0.255	0.001	0.001	11537	7754	1
beta_over_base[3]	0.141	0.086	-0.018	0.306	0.001	0.001	20051	7412	1
beta_over_base[4]	0.006	0.089	-0.166	0.167	0.001	0.001	15998	7055	1
beta_over_base[5]	0.092	0.097	-0.095	0.27	0.001	0.001	13254	7807	1

beta_over_base[6]	0.161	0.1	-0.034	0.347	0.001	0.001	12090	6934	1
beta_over_base[7]	-0.065	0.098	-0.245	0.122	0.001	0.001	12332	7423	1
beta_over_base[8]	0.021	0.098	-0.163	0.202	0.001	0.001	12476	7978	1
beta_over_base[9]	0.077	0.095	-0.094	0.259	0.001	0.001	14259	8363	1
beta_over_base[10]	0.074	0.089	-0.09	0.247	0.001	0.001	16825	7169	1
beta_over_base[11]	0.151	0.099	-0.035	0.335	0.001	0.001	13546	7935	1
beta_over_base[12]	0.145	0.085	-0.011	0.307	0.001	0.001	20537	7882	1
beta_over_base[13]	-0.013	0.097	-0.196	0.166	0.001	0.001	14485	7804	1
beta_over_base[14]	-0.045	0.095	-0.221	0.133	0.001	0.001	13561	7767	1
beta_over_base[15]	-0.051	0.094	-0.234	0.116	0.001	0.001	14118	7203	1
beta_over_base[16]	0.157	0.099	-0.034	0.337	0.001	0.001	13930	8230	1
beta_over_base[17]	-0.029	0.108	-0.245	0.161	0.001	0.001	12494	8482	1
beta_over_base[18]	0.088	0.094	-0.091	0.261	0.001	0.001	15077	7793	1
beta_over_base[19]	0.173	0.1	-0.015	0.36	0.001	0.001	13993	8208	1
beta_over_base[20]	0.175	0.09	0.005	0.346	0.001	0.001	16791	7722	1
beta_over_base[21]	0.104	0.106	-0.09	0.304	0.001	0.001	11790	8453	1
beta_over_base[22]	0.102	0.091	-0.059	0.278	0.001	0.001	16919	7542	1
beta_over_base[23]	0.059	0.096	-0.124	0.236	0.001	0.001	13886	7741	1
beta_over_base[24]	0.014	0.09	-0.164	0.173	0.001	0.001	16629	7078	1
beta_over_base[25]	-0.099	0.096	-0.288	0.074	0.001	0.001	14064	7677	1
beta_over_base[26]	-0.031	0.098	-0.214	0.155	0.001	0.001	11931	8387	1
beta_over_base[27]	-0.065	0.096	-0.247	0.115	0.001	0.001	14280	7824	1
beta_over_base[28]	-0.037	0.106	-0.237	0.164	0.001	0.001	11503	7829	1
beta_over_base[29]	0.135	0.098	-0.042	0.322	0.001	0.001	12239	7916	1
beta_over_base[30]	0.047	0.098	-0.138	0.231	0.001	0.001	12773	8375	1
beta_over_base[31]	0.118	0.095	-0.054	0.3	0.001	0.001	15114	8599	1
beta_over_base[32]	0.189	0.089	0.021	0.358	0.001	0.001	17177	7595	1
beta_over_base[33]	0.145	0.101	-0.057	0.326	0.001	0.001	13142	7927	1
beta_over_base[34]	0.206	0.091	0.043	0.384	0.001	0.001	12180	7875	1
beta_over_base[35]	0.031	0.091	-0.139	0.203	0.001	0.001	14142	7855	1
beta_over_base[36]	0.106	0.098	-0.079	0.286	0.001	0.001	14070	7909	1

beta_over_base[37]	0.186	0.092	0.017	0.359	0.001	0.001	16651	7155	1
beta_over_base[38]	0.04	0.098	-0.153	0.219	0.001	0.001	12299	8407	1
beta_over_base[39]	0.085	0.089	-0.083	0.253	0.001	0.001	16999	7453	1
beta_over_base[40]	0.142	0.099	-0.043	0.33	0.001	0.001	12719	7308	1
beta_over_base[41]	0.124	0.091	-0.045	0.293	0.001	0.001	17593	7376	1
beta_over_base[42]	0.064	0.093	-0.116	0.232	0.001	0.001	15681	7630	1
beta_over_base[43]	0.226	0.096	0.045	0.403	0.001	0.001	13770	7491	1
beta_over_base[44]	0.159	0.099	-0.029	0.343	0.001	0.001	12088	8065	1
beta_over_base[45]	0.208	0.093	0.041	0.393	0.001	0.001	13771	6174	1
beta_over_base[46]	0.174	0.097	-0.009	0.355	0.001	0.001	13433	7949	1
beta_over_base[47]	0.19	0.095	0.012	0.365	0.001	0.001	14688	7928	1
beta_over_base[48]	0.249	0.11	0.038	0.448	0.001	0.001	10227	7550	1
beta_over_base[49]	0.239	0.099	0.057	0.429	0.001	0.001	13727	8094	1
beta_over_base[50]	0.141	0.094	-0.036	0.316	0.001	0.001	15328	7898	1
beta_over_base[51]	-0.01	0.096	-0.194	0.168	0.001	0.001	11812	8391	1
beta_over_base[52]	0.069	0.094	-0.102	0.252	0.001	0.001	14397	8541	1
beta_over_base[53]	0.157	0.094	-0.022	0.331	0.001	0.001	14603	7567	1
beta_over_base[54]	0.062	0.088	-0.096	0.236	0.001	0.001	16509	7016	1
beta_over_base[55]	0.174	0.095	-0.003	0.351	0.001	0.001	15032	7854	1
beta_over_base[56]	0.142	0.096	-0.042	0.32	0.001	0.001	13415	7741	1
beta_over_base[57]	0.12	0.094	-0.052	0.301	0.001	0.001	15698	8071	1
beta_over_base[58]	0.173	0.093	-0.011	0.338	0.001	0.001	13349	7013	1
beta_over_base[59]	0.084	0.096	-0.104	0.255	0.001	0.001	11302	7891	1
beta_over_base[60]	0.185	0.098	-0.002	0.368	0.001	0.001	12731	7326	1
beta_over_base[61]	0.145	0.097	-0.04	0.325	0.001	0.001	11684	8141	1
beta_over_base[62]	0.139	0.093	-0.027	0.32	0.001	0.001	15137	8237	1
beta_over_base[63]	0.046	0.095	-0.146	0.214	0.001	0.001	14359	8129	1
beta_over_base[64]	0.009	0.1	-0.183	0.196	0.001	0.001	12020	7620	1
beta_over_base[65]	-0.045	0.108	-0.259	0.148	0.001	0.001	9887	7853	1
beta_over_base[66]	-0.004	0.093	-0.18	0.17	0.001	0.001	15782	7038	1
beta_over_base[67]	-0.003	0.096	-0.189	0.171	0.001	0.001	12058	8015	1

beta_over_base[68]	0.15	0.104	-0.05	0.342	0.001	0.001	12820	8790	1
beta_over_base[69]	0.205	0.106	0.004	0.402	0.001	0.001	12399	8888	1
beta_over_base[70]	0.151	0.093	-0.025	0.325	0.001	0.001	14959	7914	1
beta_over_base[71]	0.176	0.099	-0.011	0.359	0.001	0.001	12961	7707	1
beta_over_base[72]	0.049	0.097	-0.129	0.235	0.001	0.001	12949	7962	1
beta_over_base[73]	0.25	0.093	0.081	0.43	0.001	0.001	14032	7825	1
beta_over_base[74]	0.017	0.104	-0.182	0.207	0.001	0.001	12461	8548	1
beta_over_base[75]	0.155	0.089	-0.017	0.319	0.001	0.001	16748	7474	1
beta_over_base[76]	0.068	0.095	-0.11	0.244	0.001	0.001	16697	7811	1
beta_over_base[77]	0.2	0.098	0.012	0.384	0.001	0.001	10128	8098	1
beta_over_base[78]	0.004	0.103	-0.195	0.194	0.001	0.001	12473	8221	1
beta_over_base[79]	0.073	0.09	-0.095	0.242	0.001	0.001	18161	7938	1
beta_over_base[80]	0.13	0.092	-0.037	0.311	0.001	0.001	17393	7617	1
beta_over_base[81]	0.117	0.097	-0.064	0.299	0.001	0.001	12313	8006	1
beta_over_base[82]	0.081	0.093	-0.09	0.259	0.001	0.001	13992	7553	1
beta_over_base[83]	0.168	0.095	-0.017	0.34	0.001	0.001	16008	7661	1
beta_over_base[84]	0.226	0.105	0.025	0.416	0.001	0.001	13263	8241	1
beta_over_base[85]	0.126	0.099	-0.069	0.302	0.001	0.001	11618	7545	1
beta_over_base[86]	0.225	0.099	0.049	0.419	0.001	0.001	12623	8655	1
beta_over_base[87]	0.192	0.098	0.005	0.376	0.001	0.001	15407	7634	1
beta_over_base[88]	0.126	0.093	-0.048	0.3	0.001	0.001	15103	8555	1
beta_over_base[89]	0.225	0.098	0.04	0.406	0.001	0.001	11896	8110	1
beta_over_base[90]	0.161	0.093	-0.012	0.336	0.001	0.001	17219	7837	1
beta_over_base[91]	0.149	0.094	-0.021	0.331	0.001	0.001	14725	7461	1
beta_over_base[92]	0.063	0.093	-0.115	0.24	0.001	0.001	17571	7210	1
beta_over_base[93]	0.165	0.094	-0.016	0.338	0.001	0.001	13964	7695	1
beta_over_base[94]	0.192	0.102	0.008	0.394	0.001	0.001	13812	8751	1
beta_over_base[95]	-0.021	0.104	-0.222	0.168	0.001	0.001	12962	8081	1
beta_over_base[96]	0.085	0.104	-0.114	0.277	0.001	0.001	11384	8362	1
beta_over_base[97]	0.121	0.093	-0.054	0.294	0.001	0.001	15484	8097	1
beta_over_base[98]	0.164	0.099	-0.022	0.347	0.001	0.001	13162	7759	1

beta_over_base[99]	0.1	0.091	-0.072	0.272	0.001	0.001	16668	7139	1
beta_over_base[100]	0.129	0.097	-0.051	0.309	0.001	0.001	13880	7678	1
beta_over_base[101]	0.108	0.092	-0.067	0.279	0.001	0.001	15928	7694	1
beta_over_base[102]	0.101	0.094	-0.075	0.275	0.001	0.001	13772	7567	1
beta_over_base[103]	0.103	0.091	-0.065	0.275	0.001	0.001	12934	7558	1
beta_over_base[104]	0.129	0.096	-0.047	0.312	0.001	0.001	14770	8641	1
beta_over_base[105]	0.085	0.09	-0.08	0.26	0.001	0.001	18828	6771	1
beta_over_base[106]	0.059	0.095	-0.109	0.249	0.001	0.001	15108	7985	1
beta_over_base[107]	0.053	0.096	-0.124	0.237	0.001	0.001	14218	8339	1
beta_over_base[108]	0.113	0.101	-0.073	0.308	0.001	0.001	12415	8770	1
beta_over_base[109]	0.13	0.1	-0.053	0.322	0.001	0.001	12601	8183	1
beta_over_base[110]	0.172	0.09	0.003	0.34	0.001	0.001	17287	7299	1
beta_over_base[111]	0.131	0.09	-0.033	0.301	0.001	0.001	15448	7534	1
beta_over_base[112]	0.17	0.096	-0.014	0.341	0.001	0.001	12861	7836	1
beta_over_base[113]	0.156	0.099	-0.029	0.343	0.001	0.001	11741	7626	1
beta_over_base[114]	0.149	0.09	-0.015	0.322	0.001	0.001	16390	7214	1
beta_over_base[115]	0.076	0.09	-0.091	0.251	0.001	0.001	19657	7866	1
delta_ctrl_over[0]	-0.022	0.034	-0.082	0.044	0	0	5879	6813	1
delta_ctrl_over[1]	0.079	0.044	-0.003	0.161	0.001	0	7513	7374	1
delta_ctrl_over[2]	0.012	0.03	-0.045	0.068	0	0	6611	7884	1
eta_ctrl_late[0]	-0.077	0.018	-0.11	-0.044	0	0	5144	7150	1
eta_ctrl_late[1]	-0.061	0.025	-0.109	-0.014	0	0	5982	6997	1
eta_ctrl_late[2]	0.039	0.018	0.005	0.073	0	0	5394	6621	1
k_late_pulse	0.446	0.051	0.352	0.544	0.001	0	10015	7315	1
k_late_recovery	0.24	0.085	0.093	0.382	0.002	0.005	6819	3431	1
k_over_pulse	-1.624	0.099	-1.809	-1.438	0.001	0.001	11094	8937	1
k_over_recovery	-1.056	0.131	-1.307	-0.814	0.001	0.001	10290	7995	1
mu_alpha_late	-0.139	0.088	-0.306	0.025	0.004	0.002	616	1560	1.01
mu_alpha_over	0.566	0.073	0.429	0.702	0.001	0.001	6992	7252	1
mu_beta_late	0.103	0.024	0.06	0.148	0	0	4881	6442	1
mu_beta_over	0.108	0.045	0.024	0.193	0.001	0	6402	6687	1

phiA_over[0]	0.01	2.037	-3.872	3.75	0.013	0.024	26400	6684	1
phiA_over[1]	-0.008	0.049	-0.104	0.081	0.001	0	8035	7720	1
phiA_over[2]	0.012	0.06	-0.101	0.124	0.001	0.001	8787	8121	1
phiB_over[0]	0.344	0.102	0.157	0.539	0.001	0.001	10474	8595	1
phiB_over[1]	0.019	0.081	-0.128	0.173	0.001	0.001	10629	8696	1
phiB_over[2]	0.057	0.087	-0.107	0.219	0.001	0.001	10225	8522	1
phiB_over[3]	0.007	1.975	-3.765	3.586	0.012	0.024	25795	6921	1
phiB_over[4]	0.002	0.061	-0.11	0.115	0.001	0.001	9425	8371	1
phiB_over[5]	-0.023	0.1	-0.211	0.165	0.001	0.001	11343	8039	1
rho_late_recovery	1.778	0.736	0.365	3.18	0.01	0.011	4907	2033	1
rho_over_recovery	2.747	0.5	1.835	3.668	0.003	0.006	25802	7216	1
sd_alpha_late	0.865	0.058	0.757	0.971	0.002	0.001	1290	2877	1
sd_alpha_over	0.427	0.039	0.354	0.501	0.001	0	4015	6332	1
sd_beta_late	0.083	0.008	0.068	0.099	0	0	3540	6557	1
sd_beta_over	0.116	0.016	0.087	0.148	0	0	4494	6651	1
sigma_late	0.351	0.009	0.334	0.369	0	0	11375	8102	1
sigma_over	0.732	0.02	0.696	0.771	0	0	10267	8083	1
thetaA_late[0]	0.013	2.012	-3.763	3.759	0.012	0.026	27479	6455	1
thetaA_late[1]	-0.02	0.03	-0.075	0.037	0	0	5103	6587	1
thetaA_late[2]	0.004	0.037	-0.064	0.073	0	0	5688	6590	1
thetaB_late[0]	0.064	0.062	-0.057	0.178	0.001	0.001	7779	7530	1
thetaB_late[1]	0.059	0.049	-0.032	0.153	0.001	0	7135	7779	1
thetaB_late[2]	0.005	0.053	-0.097	0.103	0.001	0	7538	7793	1
thetaB_late[3]	0.019	1.984	-3.896	3.585	0.012	0.023	27034	7057	1
thetaB_late[4]	-0.021	0.036	-0.09	0.045	0	0	6661	7487	1
thetaB_late[5]	0.091	0.061	-0.021	0.205	0.001	0.001	8978	8276	1
z_alpha_late[0]	2.366	0.228	1.926	2.779	0.005	0.003	1830	4020	1
z_alpha_late[1]	2.457	0.232	2.051	2.92	0.005	0.003	1868	3850	1
z_alpha_late[2]	2.423	0.233	1.989	2.865	0.005	0.003	1882	3960	1
z_alpha_late[3]	0.118	0.169	-0.201	0.431	0.004	0.002	1923	4265	1
z_alpha_late[4]	-0.085	0.169	-0.419	0.217	0.004	0.002	1990	4472	1

z_alpha_late[5]	2.541	0.234	2.113	2.984	0.005	0.003	1898	3901	1
z_alpha_late[6]	1.052	0.184	0.716	1.411	0.004	0.002	2071	3871	1
z_alpha_late[7]	1.538	0.197	1.177	1.912	0.004	0.002	1999	3996	1
z_alpha_late[8]	2.543	0.235	2.103	2.985	0.006	0.003	1797	3773	1
z_alpha_late[9]	-0.859	0.178	-1.2	-0.54	0.004	0.002	1602	4142	1
z_alpha_late[10]	-0.92	0.181	-1.241	-0.563	0.004	0.002	1652	4578	1
z_alpha_late[11]	-0.313	0.172	-0.635	0.011	0.004	0.002	1688	3890	1
z_alpha_late[12]	-0.637	0.176	-0.979	-0.319	0.004	0.002	1601	3894	1
z_alpha_late[13]	0.805	0.18	0.465	1.145	0.004	0.002	2297	4416	1
z_alpha_late[14]	-0.046	0.17	-0.363	0.27	0.004	0.002	1843	4105	1
z_alpha_late[15]	0.716	0.179	0.374	1.048	0.004	0.002	1949	4364	1
z_alpha_late[16]	-0.01	0.167	-0.333	0.291	0.004	0.002	1721	4196	1
z_alpha_late[17]	-0.367	0.173	-0.678	-0.028	0.004	0.002	1688	3842	1
z_alpha_late[18]	0.866	0.177	0.538	1.195	0.004	0.002	2025	4177	1
z_alpha_late[19]	0.334	0.173	0.022	0.677	0.004	0.002	2084	4175	1
z_alpha_late[20]	0.143	0.171	-0.185	0.461	0.004	0.002	1911	4496	1
z_alpha_late[21]	-0.379	0.169	-0.7	-0.067	0.004	0.002	1607	3942	1
z_alpha_late[22]	-0.614	0.177	-0.961	-0.294	0.004	0.002	1828	3722	1
z_alpha_late[23]	-0.181	0.17	-0.487	0.147	0.004	0.002	1933	4100	1
z_alpha_late[24]	-1.086	0.185	-1.437	-0.749	0.005	0.002	1486	3950	1
z_alpha_late[25]	-0.464	0.175	-0.79	-0.132	0.004	0.002	1700	4140	1
z_alpha_late[26]	0.832	0.177	0.499	1.164	0.004	0.002	1976	4175	1
z_alpha_late[27]	-0.984	0.182	-1.332	-0.651	0.005	0.002	1469	3465	1
z_alpha_late[28]	-0.661	0.177	-1.001	-0.345	0.004	0.002	1709	4244	1
z_alpha_late[29]	-1.288	0.193	-1.652	-0.927	0.005	0.002	1431	3841	1
z_alpha_late[30]	0.381	0.172	0.057	0.705	0.004	0.002	1904	3822	1
z_alpha_late[31]	-0.698	0.179	-1.041	-0.374	0.004	0.002	1726	4151	1
z_alpha_late[32]	0.203	0.173	-0.121	0.521	0.004	0.002	2032	3967	1
z_alpha_late[33]	-0.215	0.17	-0.529	0.105	0.004	0.002	1569	4361	1
z_alpha_late[34]	-0.175	0.173	-0.521	0.132	0.004	0.002	1789	3927	1
z_alpha_late[35]	0.2	0.171	-0.105	0.531	0.004	0.002	1956	4281	1

z_alpha_late[36]	0.564	0.173	0.239	0.888	0.004	0.002	1926	4343	1
z_alpha_late[37]	-0.303	0.173	-0.628	0.017	0.004	0.002	1683	4004	1
z_alpha_late[38]	0.179	0.169	-0.144	0.487	0.004	0.002	2028	4676	1
z_alpha_late[39]	-0.829	0.178	-1.152	-0.486	0.005	0.002	1515	3811	1
z_alpha_late[40]	0.185	0.171	-0.152	0.491	0.004	0.002	1917	3850	1
z_alpha_late[41]	0.202	0.171	-0.135	0.509	0.004	0.002	1821	4130	1
z_alpha_late[42]	0.99	0.181	0.656	1.341	0.004	0.002	2029	4072	1
z_alpha_late[43]	-0.429	0.174	-0.733	-0.083	0.004	0.002	1660	4464	1
z_alpha_late[44]	0.308	0.172	-0.007	0.646	0.004	0.002	2016	4138	1
z_alpha_late[45]	-0.491	0.177	-0.831	-0.167	0.004	0.002	1784	4160	1
z_alpha_late[46]	0.868	0.18	0.542	1.22	0.004	0.002	2132	4089	1
z_alpha_late[47]	0.002	0.17	-0.328	0.314	0.004	0.002	1906	4053	1
z_alpha_late[48]	-0.382	0.173	-0.717	-0.074	0.004	0.002	1679	4242	1
z_alpha_late[49]	-0.284	0.17	-0.592	0.034	0.004	0.002	1779	3566	1
z_alpha_late[50]	2.333	0.227	1.926	2.774	0.005	0.003	1834	3641	1
z_alpha_late[51]	1.089	0.183	0.742	1.435	0.004	0.002	1922	4672	1
z_alpha_late[52]	0.65	0.178	0.33	0.996	0.004	0.002	1973	4370	1
z_alpha_late[53]	0.803	0.178	0.476	1.143	0.004	0.002	1985	3807	1
z_alpha_late[54]	0.581	0.176	0.251	0.911	0.004	0.002	2049	4215	1
z_alpha_late[55]	-1.336	0.192	-1.707	-0.986	0.005	0.002	1420	3502	1
z_alpha_late[56]	-0.292	0.171	-0.596	0.047	0.004	0.002	1761	4022	1
z_alpha_late[57]	-0.16	0.174	-0.485	0.163	0.004	0.002	1945	4445	1
z_alpha_late[58]	-0.275	0.171	-0.605	0.04	0.004	0.002	1772	4199	1
z_alpha_late[59]	1.611	0.196	1.227	1.968	0.004	0.002	1969	3868	1
z_alpha_late[60]	1.726	0.203	1.34	2.108	0.005	0.002	2013	3321	1
z_alpha_late[61]	2.6	0.236	2.157	3.041	0.006	0.003	1769	3338	1
z_alpha_late[62]	0.446	0.17	0.108	0.747	0.004	0.002	1942	3864	1
z_alpha_late[63]	-1.374	0.193	-1.729	-1.016	0.005	0.002	1350	4017	1
z_alpha_late[64]	-0.157	0.172	-0.48	0.167	0.004	0.002	1844	3730	1
z_alpha_late[65]	0.472	0.173	0.144	0.795	0.004	0.002	2061	3538	1
z_alpha_late[66]	1.546	0.199	1.184	1.931	0.004	0.002	2005	4056	1

z_alpha_late[67]	-0.628	0.172	-0.944	-0.297	0.004	0.002	1596	3609	1
z_alpha_late[68]	0.095	0.172	-0.223	0.421	0.004	0.002	1950	3586	1
z_alpha_late[69]	0.127	0.172	-0.199	0.453	0.004	0.002	1790	4131	1
z_alpha_late[70]	-0.373	0.174	-0.699	-0.042	0.004	0.002	1719	3395	1
z_alpha_late[71]	-0.32	0.171	-0.64	-0.002	0.004	0.002	1821	4589	1
z_alpha_late[72]	2.16	0.22	1.761	2.581	0.005	0.003	1877	3841	1
z_alpha_late[73]	-0.327	0.175	-0.646	0.004	0.004	0.002	1761	3934	1
z_alpha_late[74]	-0.471	0.175	-0.8	-0.141	0.004	0.002	1858	4303	1
z_alpha_late[75]	-1.216	0.188	-1.582	-0.871	0.005	0.002	1425	3368	1
z_alpha_late[76]	-1.194	0.185	-1.534	-0.84	0.005	0.002	1415	3866	1
z_alpha_late[77]	-1.212	0.188	-1.565	-0.856	0.005	0.002	1385	3218	1
z_alpha_late[78]	1.217	0.188	0.874	1.584	0.004	0.002	1996	3467	1
z_alpha_late[79]	0.332	0.172	0.017	0.649	0.004	0.002	1883	3718	1
z_alpha_late[80]	-0.913	0.18	-1.253	-0.58	0.005	0.002	1586	4001	1
z_alpha_late[81]	-0.05	0.171	-0.376	0.265	0.004	0.002	1924	3995	1
z_alpha_late[82]	0.293	0.171	-0.015	0.627	0.004	0.002	2038	3870	1
z_alpha_late[83]	0.319	0.173	0.006	0.655	0.004	0.002	1980	4134	1
z_alpha_late[84]	-0.744	0.18	-1.092	-0.417	0.004	0.002	1634	4264	1
z_alpha_late[85]	-1.166	0.186	-1.525	-0.832	0.005	0.002	1488	4240	1
z_alpha_late[86]	-1.903	0.212	-2.303	-1.5	0.006	0.002	1415	3753	1
z_alpha_late[87]	-0.27	0.172	-0.597	0.051	0.004	0.002	1732	3715	1
z_alpha_late[88]	-0.461	0.175	-0.791	-0.141	0.004	0.002	1671	3608	1
z_alpha_late[89]	-1.285	0.192	-1.646	-0.922	0.005	0.002	1406	3540	1
z_alpha_late[90]	-0.974	0.181	-1.312	-0.632	0.005	0.002	1549	3725	1
z_alpha_late[91]	-1.416	0.195	-1.779	-1.051	0.005	0.002	1478	3933	1
z_alpha_late[92]	-0.783	0.178	-1.118	-0.448	0.004	0.002	1753	4596	1
z_alpha_late[93]	-0.983	0.183	-1.322	-0.636	0.005	0.002	1600	3689	1
z_alpha_late[94]	-1.24	0.187	-1.58	-0.882	0.005	0.002	1398	3542	1
z_alpha_late[95]	-0.047	0.17	-0.375	0.26	0.004	0.002	1614	3886	1
z_alpha_late[96]	0.273	0.169	-0.06	0.58	0.004	0.002	1884	3993	1
z_alpha_late[97]	0.065	0.172	-0.25	0.385	0.004	0.002	1964	3693	1

z_alpha_late[98]	1.062	0.181	0.74	1.419	0.004	0.002	2011	4098	1
z_alpha_late[99]	-0.201	0.171	-0.515	0.133	0.004	0.002	1701	4174	1
z_alpha_late[100]	-0.864	0.177	-1.205	-0.542	0.005	0.002	1480	3679	1
z_alpha_late[101]	-1.011	0.184	-1.358	-0.666	0.005	0.002	1596	3386	1
z_alpha_late[102]	-0.427	0.171	-0.746	-0.1	0.004	0.002	1609	4317	1
z_alpha_late[103]	-0.873	0.179	-1.202	-0.538	0.005	0.002	1455	4144	1
z_alpha_late[104]	0.397	0.173	0.088	0.737	0.004	0.002	2139	4193	1
z_alpha_late[105]	-1.299	0.19	-1.645	-0.939	0.005	0.002	1536	3673	1
z_alpha_late[106]	-0.937	0.181	-1.277	-0.601	0.005	0.002	1299	3675	1
z_alpha_late[107]	0.848	0.178	0.52	1.186	0.004	0.002	2053	4000	1
z_alpha_late[108]	0.076	0.169	-0.248	0.384	0.004	0.002	1803	3916	1
z_alpha_late[109]	-0.763	0.178	-1.097	-0.429	0.005	0.002	1554	3975	1
z_alpha_late[110]	-0.745	0.176	-1.087	-0.427	0.004	0.002	1628	2937	1
z_alpha_late[111]	0.05	0.17	-0.272	0.372	0.004	0.002	1855	4159	1
z_alpha_late[112]	0.01	0.171	-0.302	0.337	0.004	0.002	1841	4209	1
z_alpha_late[113]	-1.129	0.19	-1.487	-0.78	0.005	0.002	1453	4084	1
z_alpha_late[114]	-0.239	0.171	-0.544	0.096	0.004	0.002	1903	4360	1
z_alpha_late[115]	-0.837	0.178	-1.166	-0.501	0.005	0.002	1450	3864	1
z_alpha_over[0]	0.815	0.53	-0.175	1.815	0.003	0.007	24662	6452	1
z_alpha_over[1]	0.886	0.516	-0.072	1.858	0.003	0.006	22393	7228	1
z_alpha_over[2]	-0.218	0.534	-1.221	0.769	0.003	0.007	24898	7092	1
z_alpha_over[3]	0.369	0.53	-0.653	1.326	0.003	0.006	25344	6839	1
z_alpha_over[4]	0.625	0.536	-0.327	1.7	0.004	0.007	22122	7436	1
z_alpha_over[5]	-0.057	0.531	-1.048	0.956	0.004	0.006	21671	6303	1
z_alpha_over[6]	0.083	0.527	-0.887	1.115	0.003	0.007	26299	6567	1
z_alpha_over[7]	-0.494	0.532	-1.5	0.483	0.003	0.006	23732	6921	1
z_alpha_over[8]	0.846	0.532	-0.192	1.799	0.003	0.006	24906	6902	1
z_alpha_over[9]	0.219	0.535	-0.742	1.262	0.003	0.007	24023	6784	1
z_alpha_over[10]	0.206	0.523	-0.817	1.142	0.003	0.007	22615	6605	1
z_alpha_over[11]	0.049	0.516	-0.924	1.023	0.003	0.006	23922	7308	1
z_alpha_over[12]	0.104	0.519	-0.914	1.049	0.003	0.006	22802	6848	1

z_alpha_over[13]	-0.274	0.516	-1.193	0.774	0.003	0.006	24076	6348	1
z_alpha_over[14]	0.381	0.535	-0.588	1.401	0.003	0.007	25335	6136	1
z_alpha_over[15]	-1.394	0.536	-2.407	-0.388	0.004	0.006	21192	6348	1
z_alpha_over[16]	0.535	0.529	-0.411	1.565	0.003	0.006	27339	7009	1
z_alpha_over[17]	-1.142	0.53	-2.094	-0.107	0.003	0.006	25162	7389	1
z_alpha_over[18]	-0.55	0.53	-1.516	0.482	0.003	0.006	23448	6790	1
z_alpha_over[19]	-0.041	0.525	-1.018	0.949	0.003	0.007	23412	6666	1
z_alpha_over[20]	2.106	0.539	1.055	3.092	0.004	0.006	21782	7604	1
z_alpha_over[21]	-0.222	0.52	-1.19	0.746	0.003	0.006	25219	7419	1
z_alpha_over[22]	-0.824	0.533	-1.845	0.176	0.003	0.007	26185	7096	1
z_alpha_over[23]	1.585	0.534	0.51	2.529	0.004	0.007	22710	6919	1
z_alpha_over[24]	-0.672	0.533	-1.677	0.31	0.003	0.007	24116	6971	1
z_alpha_over[25]	-0.648	0.532	-1.673	0.313	0.004	0.007	23007	6799	1
z_alpha_over[26]	0.443	0.533	-0.544	1.46	0.003	0.006	25887	7491	1
z_alpha_over[27]	-0.238	0.537	-1.234	0.782	0.003	0.006	24356	6973	1
z_alpha_over[28]	0.661	0.533	-0.331	1.671	0.004	0.007	22009	6435	1
z_alpha_over[29]	0.847	0.526	-0.138	1.847	0.003	0.006	24111	7072	1
z_alpha_over[30]	0.235	0.526	-0.749	1.211	0.003	0.007	22842	6396	1
z_alpha_over[31]	-2.067	0.533	-3.035	-1.027	0.003	0.006	23854	7267	1
z_alpha_over[32]	1.215	0.534	0.231	2.223	0.004	0.006	22887	7301	1
z_alpha_over[33]	-0.244	0.518	-1.242	0.705	0.003	0.006	23365	6406	1
z_alpha_over[34]	-1.008	0.527	-2.016	-0.061	0.003	0.006	23388	6839	1
z_alpha_over[35]	-0.195	0.539	-1.182	0.851	0.003	0.007	25351	7246	1
z_alpha_over[36]	0.053	0.533	-0.96	1.039	0.003	0.007	23867	6512	1
z_alpha_over[37]	-0.242	0.531	-1.234	0.777	0.003	0.007	24225	6532	1
z_alpha_over[38]	0.068	0.529	-0.91	1.066	0.003	0.006	24420	7065	1
z_alpha_over[39]	-0.955	0.537	-1.943	0.076	0.004	0.006	22452	7041	1
z_alpha_over[40]	0.668	0.525	-0.33	1.664	0.003	0.006	23647	7200	1
z_alpha_over[41]	0.434	0.518	-0.521	1.447	0.003	0.006	22949	6585	1
z_alpha_over[42]	0.026	0.524	-0.963	0.98	0.003	0.006	24518	7081	1
z_alpha_over[43]	-0.515	0.53	-1.53	0.455	0.003	0.007	23425	6218	1

z_alpha_over[44]	0.015	0.532	-0.988	0.993	0.003	0.006	27728	6792	1
z_alpha_over[45]	0.033	0.526	-0.93	1.003	0.003	0.006	24188	6587	1
z_alpha_over[46]	-0.538	0.522	-1.512	0.44	0.003	0.006	24097	6906	1
z_alpha_over[47]	-1.064	0.527	-2.073	-0.076	0.004	0.006	21998	6805	1
z_alpha_over[48]	0.431	0.519	-0.539	1.394	0.003	0.006	25036	7018	1
z_alpha_over[49]	1.37	0.528	0.384	2.34	0.003	0.006	25047	7091	1
z_alpha_over[50]	-1.231	0.534	-2.248	-0.229	0.003	0.007	23894	7072	1
z_alpha_over[51]	-1.896	0.541	-2.928	-0.885	0.004	0.007	21811	6365	1
z_alpha_over[52]	-0.06	0.525	-1.026	0.95	0.003	0.006	24379	7178	1
z_alpha_over[53]	0.175	0.533	-0.874	1.128	0.003	0.007	25759	6994	1
z_alpha_over[54]	-1.727	0.543	-2.745	-0.734	0.004	0.007	22052	7484	1
z_alpha_over[55]	-0.899	0.54	-1.968	0.072	0.004	0.006	23396	6949	1
z_alpha_over[56]	0.385	0.533	-0.573	1.398	0.003	0.006	25023	7187	1
z_alpha_over[57]	0.002	0.531	-1.057	0.95	0.004	0.007	22641	6956	1
z_alpha_over[58]	0.326	0.535	-0.658	1.36	0.003	0.007	25630	6628	1
z_alpha_over[59]	0.325	0.524	-0.672	1.266	0.003	0.006	22354	6890	1
z_alpha_over[60]	0.613	0.533	-0.364	1.628	0.003	0.006	23389	6767	1
z_alpha_over[61]	0.651	0.526	-0.344	1.629	0.003	0.006	23678	7277	1
z_alpha_over[62]	1.225	0.519	0.209	2.158	0.004	0.006	21607	6917	1
z_alpha_over[63]	-1.419	0.537	-2.397	-0.391	0.003	0.006	23910	7255	1
z_alpha_over[64]	-1.331	0.529	-2.391	-0.406	0.003	0.006	24829	6985	1
z_alpha_over[65]	-0.717	0.536	-1.697	0.321	0.003	0.007	23895	6606	1
z_alpha_over[66]	-0.52	0.527	-1.481	0.495	0.003	0.006	24137	6808	1
z_alpha_over[67]	-2.061	0.521	-3.077	-1.118	0.004	0.006	21214	7275	1
z_alpha_over[68]	-1.976	0.526	-3.024	-1.028	0.004	0.006	20087	6551	1
z_alpha_over[69]	-0.51	0.529	-1.512	0.476	0.003	0.007	25268	6806	1
z_alpha_over[70]	-0.249	0.533	-1.237	0.768	0.003	0.007	25093	6670	1
z_alpha_over[71]	-1.394	0.548	-2.397	-0.36	0.004	0.007	22501	6373	1
z_alpha_over[72]	0.724	0.529	-0.294	1.692	0.003	0.007	24640	6484	1
z_alpha_over[73]	1.284	0.529	0.286	2.26	0.004	0.006	22257	6962	1
z_alpha_over[74]	0.717	0.523	-0.26	1.696	0.003	0.006	23949	7462	1

z_alpha_over[75]	-0.522	0.524	-1.471	0.472	0.003	0.006	24057	7405	1
z_alpha_over[76]	0.791	0.537	-0.239	1.777	0.004	0.007	22344	5601	1
z_alpha_over[77]	0.268	0.533	-0.737	1.242	0.004	0.006	21651	7323	1
z_alpha_over[78]	-0.604	0.523	-1.576	0.373	0.003	0.007	24550	6493	1
z_alpha_over[79]	0.484	0.523	-0.528	1.441	0.003	0.006	22418	6292	1
z_alpha_over[80]	-0.826	0.526	-1.823	0.134	0.003	0.006	23556	7117	1
z_alpha_over[81]	0.441	0.519	-0.489	1.446	0.003	0.006	24459	6616	1
z_alpha_over[82]	-0.226	0.52	-1.2	0.768	0.003	0.006	22938	6558	1
z_alpha_over[83]	-0.141	0.531	-1.119	0.866	0.003	0.006	26307	7218	1
z_alpha_over[84]	1.584	0.523	0.626	2.585	0.004	0.006	20331	7226	1
z_alpha_over[85]	0.601	0.515	-0.392	1.557	0.003	0.006	24031	7665	1
z_alpha_over[86]	0.428	0.528	-0.532	1.433	0.003	0.007	23834	6533	1
z_alpha_over[87]	1.245	0.532	0.244	2.268	0.004	0.007	22158	5924	1
z_alpha_over[88]	1.209	0.535	0.241	2.25	0.003	0.006	23448	7138	1
z_alpha_over[89]	1.564	0.536	0.585	2.593	0.004	0.006	20589	6686	1
z_alpha_over[90]	-0.459	0.526	-1.405	0.53	0.003	0.007	25435	7437	1
z_alpha_over[91]	1.075	0.522	0.112	2.067	0.003	0.006	23881	6926	1
z_alpha_over[92]	0.24	0.522	-0.713	1.23	0.003	0.006	23198	7103	1
z_alpha_over[93]	1.157	0.529	0.18	2.177	0.004	0.007	22011	6593	1
z_alpha_over[94]	1.731	0.535	0.742	2.739	0.004	0.006	22363	7082	1
z_alpha_over[95]	-0.738	0.536	-1.759	0.266	0.003	0.006	24760	6979	1
z_alpha_over[96]	-0.226	0.521	-1.209	0.755	0.003	0.006	23788	6591	1
z_alpha_over[97]	0.508	0.525	-0.461	1.51	0.003	0.006	22568	7179	1
z_alpha_over[98]	0.246	0.52	-0.747	1.203	0.003	0.007	22538	6338	1
z_alpha_over[99]	-0.279	0.536	-1.282	0.712	0.003	0.006	26780	6936	1
z_alpha_over[100]	0.825	0.524	-0.18	1.805	0.003	0.006	25956	6954	1
z_alpha_over[101]	0.56	0.532	-0.412	1.583	0.004	0.007	21587	6803	1
z_alpha_over[102]	-0.27	0.514	-1.255	0.677	0.003	0.006	22048	7068	1
z_alpha_over[103]	-0.553	0.525	-1.514	0.455	0.003	0.006	24671	6747	1
z_alpha_over[104]	-0.167	0.526	-1.139	0.833	0.003	0.007	25338	6058	1
z_alpha_over[105]	-0.427	0.527	-1.453	0.551	0.003	0.007	24475	6827	1

z_alpha_over[106]	-0.38	0.532	-1.37	0.616	0.003	0.006	23331	7057	1
z_alpha_over[107]	-1.477	0.526	-2.455	-0.49	0.003	0.007	24801	7018	1
z_alpha_over[108]	-0.426	0.523	-1.435	0.546	0.003	0.006	23297	6864	1
z_alpha_over[109]	1.072	0.527	0.101	2.051	0.004	0.007	22578	6405	1
z_alpha_over[110]	-0.803	0.53	-1.809	0.191	0.003	0.007	27903	6550	1
z_alpha_over[111]	0.029	0.532	-0.954	1.026	0.003	0.006	25341	6034	1
z_alpha_over[112]	0.187	0.522	-0.773	1.179	0.003	0.006	23230	6968	1
z_alpha_over[113]	0.21	0.529	-0.794	1.174	0.003	0.006	24676	6563	1
z_alpha_over[114]	-0.303	0.518	-1.236	0.691	0.003	0.006	26850	6833	1
z_alpha_over[115]	0.23	0.527	-0.743	1.243	0.004	0.006	22636	6789	1
z_beta_late[0]	-1.115	0.639	-2.294	0.12	0.006	0.006	13043	8275	1
z_beta_late[1]	0.704	0.62	-0.428	1.891	0.006	0.006	12097	7938	1
z_beta_late[2]	1.003	0.616	-0.132	2.187	0.005	0.006	14908	7698	1
z_beta_late[3]	-1.133	0.565	-2.192	-0.066	0.005	0.006	14710	7312	1
z_beta_late[4]	-0.777	0.582	-1.87	0.295	0.004	0.007	16719	7486	1
z_beta_late[5]	0.563	0.569	-0.535	1.584	0.004	0.007	19870	7351	1
z_beta_late[6]	1.065	0.581	-0.035	2.145	0.004	0.006	17325	7536	1
z_beta_late[7]	-0.479	0.572	-1.59	0.569	0.004	0.007	18497	7221	1
z_beta_late[8]	1.061	0.595	-0.027	2.212	0.005	0.006	15629	7728	1
z_beta_late[9]	-0.388	0.623	-1.508	0.822	0.005	0.006	15138	7994	1
z_beta_late[10]	-0.019	0.566	-1.049	1.077	0.004	0.006	19588	7060	1
z_beta_late[11]	-0.251	0.618	-1.401	0.911	0.005	0.006	14515	8197	1
z_beta_late[12]	-1.47	0.586	-2.559	-0.378	0.005	0.006	14275	7502	1
z_beta_late[13]	0.055	0.613	-1.121	1.202	0.005	0.006	12646	6838	1
z_beta_late[14]	0.075	0.597	-1.059	1.184	0.005	0.007	15149	7099	1
z_beta_late[15]	-1.084	0.58	-2.205	-0.018	0.005	0.006	16321	8042	1
z_beta_late[16]	-0.729	0.674	-1.991	0.52	0.006	0.006	12400	8145	1
z_beta_late[17]	0.066	0.739	-1.293	1.464	0.007	0.006	12454	8387	1
z_beta_late[18]	-0.355	0.605	-1.522	0.742	0.005	0.006	13785	7736	1
z_beta_late[19]	-0.661	0.668	-1.93	0.571	0.006	0.006	12822	8072	1
z_beta_late[20]	-0.121	0.603	-1.236	1.026	0.005	0.007	15902	7330	1

z_beta_late[21]	2.133	0.68	0.835	3.358	0.006	0.006	11730	8555	1
z_beta_late[22]	-1.138	0.578	-2.199	-0.044	0.004	0.007	17657	7187	1
z_beta_late[23]	-0.659	0.609	-1.807	0.491	0.005	0.006	15785	7658	1
z_beta_late[24]	-0.053	0.578	-1.128	1.05	0.004	0.007	19046	7245	1
z_beta_late[25]	-0.3	0.566	-1.374	0.765	0.004	0.006	17337	7124	1
z_beta_late[26]	1.33	0.59	0.216	2.425	0.005	0.007	15842	7048	1
z_beta_late[27]	-1.104	0.571	-2.149	-0.023	0.004	0.006	17949	7450	1
z_beta_late[28]	-0.462	0.693	-1.746	0.853	0.006	0.006	11580	8280	1
z_beta_late[29]	-0.042	0.623	-1.202	1.106	0.005	0.006	15253	8216	1
z_beta_late[30]	0.581	0.604	-0.565	1.708	0.005	0.006	14245	7902	1
z_beta_late[31]	-0.315	0.616	-1.474	0.847	0.005	0.006	13023	7436	1
z_beta_late[32]	-0.549	0.575	-1.61	0.553	0.004	0.007	19358	7161	1
z_beta_late[33]	-0.306	0.659	-1.556	0.917	0.006	0.006	12083	8229	1
z_beta_late[34]	0.732	0.588	-0.322	1.867	0.004	0.007	18810	6966	1
z_beta_late[35]	0.776	0.588	-0.262	1.947	0.004	0.007	17698	7061	1
z_beta_late[36]	0.22	0.61	-0.931	1.363	0.005	0.006	13534	7695	1
z_beta_late[37]	0.091	0.573	-0.982	1.167	0.004	0.006	17723	7492	1
z_beta_late[38]	-0.382	0.628	-1.473	0.879	0.005	0.006	13868	7445	1
z_beta_late[39]	-0.553	0.57	-1.593	0.55	0.004	0.006	19317	7728	1
z_beta_late[40]	0.761	0.606	-0.388	1.885	0.005	0.006	16132	7984	1
z_beta_late[41]	0.197	0.595	-0.884	1.351	0.005	0.006	15731	7350	1
z_beta_late[42]	-1.092	0.57	-2.167	-0.012	0.004	0.007	20517	7129	1
z_beta_late[43]	0.352	0.566	-0.746	1.367	0.004	0.006	19899	7749	1
z_beta_late[44]	1.085	0.581	0.033	2.218	0.004	0.006	18205	7988	1
z_beta_late[45]	-0.703	0.575	-1.796	0.362	0.004	0.007	22003	6632	1
z_beta_late[46]	-0.199	0.62	-1.408	0.924	0.005	0.006	15326	8280	1
z_beta_late[47]	-0.039	0.559	-1.071	1.019	0.004	0.007	20618	7097	1
z_beta_late[48]	0.673	0.734	-0.661	2.126	0.007	0.007	11309	8274	1
z_beta_late[49]	0.087	0.563	-0.973	1.147	0.004	0.006	19046	7251	1
z_beta_late[50]	-0.786	0.57	-1.85	0.301	0.004	0.006	20144	7268	1
z_beta_late[51]	0.975	0.565	-0.117	2.027	0.004	0.007	21610	6311	1

z_beta_late[52]	-0.301	0.571	-1.363	0.794	0.004	0.007	18903	6252	1
z_beta_late[53]	-0.373	0.576	-1.448	0.715	0.004	0.007	19631	7433	1
z_beta_late[54]	0.083	0.569	-0.999	1.138	0.004	0.007	16823	6793	1
z_beta_late[55]	-1.176	0.577	-2.247	-0.08	0.004	0.007	17836	6545	1
z_beta_late[56]	-0.205	0.559	-1.301	0.8	0.004	0.006	17428	7553	1
z_beta_late[57]	-0.819	0.572	-1.865	0.293	0.004	0.006	18485	6931	1
z_beta_late[58]	0.963	0.565	-0.075	2.051	0.005	0.006	15399	7472	1
z_beta_late[59]	1.156	0.557	0.12	2.188	0.004	0.007	20587	6167	1
z_beta_late[60]	0.301	0.575	-0.844	1.351	0.004	0.007	19459	6101	1
z_beta_late[61]	1.167	0.591	0.041	2.231	0.004	0.007	18485	7296	1
z_beta_late[62]	-0.257	0.571	-1.335	0.807	0.004	0.007	24159	6926	1
z_beta_late[63]	0.575	0.574	-0.504	1.66	0.004	0.007	21725	6765	1
z_beta_late[64]	-0.19	0.691	-1.496	1.095	0.006	0.006	12584	8459	1
z_beta_late[65]	3.357	0.583	2.294	4.458	0.005	0.006	16433	8225	1
z_beta_late[66]	-2.33	0.589	-3.47	-1.246	0.005	0.006	14808	8038	1
z_beta_late[67]	-0.119	0.597	-1.241	1.007	0.005	0.007	16661	7217	1
z_beta_late[68]	-0.208	0.728	-1.588	1.134	0.006	0.007	12678	8236	1
z_beta_late[69]	0.116	0.724	-1.233	1.49	0.007	0.007	12244	8405	1
z_beta_late[70]	-1.18	0.566	-2.3	-0.164	0.004	0.006	18832	7391	1
z_beta_late[71]	0.295	0.59	-0.839	1.393	0.004	0.006	17487	7721	1
z_beta_late[72]	0.302	0.569	-0.816	1.308	0.004	0.007	22816	7014	1
z_beta_late[73]	0.282	0.571	-0.767	1.366	0.004	0.006	19401	7831	1
z_beta_late[74]	0.251	0.711	-1.062	1.59	0.007	0.006	11821	7531	1
z_beta_late[75]	-0.042	0.57	-1.14	1.008	0.005	0.006	15885	7517	1
z_beta_late[76]	0.046	0.567	-0.99	1.12	0.004	0.006	19922	8285	1
z_beta_late[77]	1.51	0.611	0.369	2.688	0.005	0.006	13480	7992	1
z_beta_late[78]	1.656	0.654	0.441	2.882	0.005	0.006	15442	8820	1
z_beta_late[79]	-0.702	0.599	-1.835	0.431	0.005	0.006	12752	7655	1
z_beta_late[80]	-1.569	0.585	-2.691	-0.503	0.004	0.007	18923	7054	1
z_beta_late[81]	0.811	0.568	-0.256	1.879	0.004	0.007	20571	6845	1
z_beta_late[82]	-0.821	0.588	-1.916	0.312	0.004	0.006	20050	7426	1

z_beta_late[83]	0.377	0.595	-0.699	1.545	0.004	0.006	19144	7979	1
z_beta_late[84]	0.03	0.69	-1.26	1.315	0.006	0.006	12782	8524	1
z_beta_late[85]	0.246	0.617	-0.915	1.385	0.005	0.006	14313	7841	1
z_beta_late[86]	1.088	0.571	-0.029	2.109	0.004	0.007	21772	7385	1
z_beta_late[87]	0.186	0.632	-1.055	1.321	0.005	0.006	15365	8374	1
z_beta_late[88]	0.544	0.595	-0.558	1.679	0.004	0.007	19507	7464	1
z_beta_late[89]	-0.214	0.602	-1.34	0.907	0.005	0.006	15546	8357	1
z_beta_late[90]	-0.566	0.613	-1.665	0.612	0.005	0.006	12669	7354	1
z_beta_late[91]	-0.476	0.576	-1.572	0.583	0.004	0.007	21580	7193	1
z_beta_late[92]	-0.878	0.617	-2.043	0.285	0.005	0.006	16110	7828	1
z_beta_late[93]	0.343	0.566	-0.748	1.396	0.004	0.007	22619	7114	1
z_beta_late[94]	-0.508	0.705	-1.754	0.874	0.006	0.007	13804	8878	1
z_beta_late[95]	-0.422	0.73	-1.734	0.986	0.006	0.006	13668	9197	1
z_beta_late[96]	0.585	0.693	-0.731	1.859	0.006	0.006	12910	8352	1
z_beta_late[97]	0.319	0.596	-0.804	1.427	0.005	0.006	12689	6953	1
z_beta_late[98]	0.68	0.598	-0.484	1.759	0.005	0.006	16435	8146	1
z_beta_late[99]	-0.38	0.57	-1.47	0.645	0.004	0.007	18195	7317	1
z_beta_late[100]	-0.004	0.603	-1.118	1.132	0.005	0.006	15197	8345	1
z_beta_late[101]	0.137	0.572	-0.956	1.172	0.004	0.006	18320	7674	1
z_beta_late[102]	0.659	0.583	-0.417	1.765	0.004	0.006	17221	7714	1
z_beta_late[103]	0.772	0.563	-0.296	1.842	0.004	0.007	18675	7059	1
z_beta_late[104]	-0.329	0.605	-1.498	0.771	0.005	0.006	16008	8391	1
z_beta_late[105]	-0.634	0.575	-1.721	0.456	0.005	0.006	15615	7001	1
z_beta_late[106]	-0.727	0.612	-1.869	0.393	0.005	0.007	17360	8073	1
z_beta_late[107]	-0.022	0.62	-1.153	1.173	0.005	0.006	15535	7901	1
z_beta_late[108]	-0.445	0.693	-1.787	0.81	0.006	0.006	12773	8172	1
z_beta_late[109]	-0.274	0.636	-1.436	0.936	0.005	0.007	15756	7587	1
z_beta_late[110]	-0.716	0.581	-1.831	0.337	0.004	0.007	18115	7094	1
z_beta_late[111]	0.239	0.566	-0.811	1.316	0.004	0.006	17838	7069	1
z_beta_late[112]	0.368	0.602	-0.714	1.54	0.005	0.006	14149	8331	1
z_beta_late[113]	0.464	0.629	-0.715	1.641	0.005	0.006	17404	7997	1

z_beta_late[114]	-0.227	0.581	-1.301	0.901	0.004	0.007	22170	7152	1
z_beta_late[115]	-0.158	0.567	-1.192	0.933	0.004	0.007	17869	6947	1
z_beta_over[0]	1.076	0.748	-0.322	2.511	0.006	0.008	16671	8046	1
z_beta_over[1]	0.561	0.738	-0.806	1.969	0.005	0.008	18391	7630	1
z_beta_over[2]	-0.289	0.74	-1.689	1.101	0.005	0.008	19587	8153	1
z_beta_over[3]	0.277	0.72	-1.06	1.62	0.005	0.008	21696	7405	1
z_beta_over[4]	-0.883	0.72	-2.152	0.58	0.005	0.009	21356	6866	1
z_beta_over[5]	-0.139	0.721	-1.496	1.198	0.005	0.008	23881	7503	1
z_beta_over[6]	0.45	0.731	-0.873	1.881	0.005	0.009	22679	6176	1
z_beta_over[7]	-1.487	0.724	-2.89	-0.159	0.005	0.009	22275	6664	1
z_beta_over[8]	-0.754	0.735	-2.15	0.602	0.005	0.008	22882	7628	1
z_beta_over[9]	-0.269	0.743	-1.704	1.101	0.005	0.008	19854	7455	1
z_beta_over[10]	-0.297	0.709	-1.684	0.988	0.005	0.009	24339	6929	1
z_beta_over[11]	0.367	0.748	-1.037	1.793	0.005	0.009	22391	6911	1
z_beta_over[12]	0.316	0.727	-1.058	1.667	0.005	0.008	20741	7066	1
z_beta_over[13]	-1.046	0.744	-2.465	0.298	0.005	0.008	20384	7080	1
z_beta_over[14]	-1.316	0.742	-2.645	0.161	0.005	0.009	20759	7151	1
z_beta_over[15]	-1.368	0.728	-2.676	0.05	0.005	0.009	19076	6789	1
z_beta_over[16]	0.415	0.772	-1.045	1.833	0.006	0.008	18077	7967	1
z_beta_over[17]	-1.181	0.82	-2.708	0.383	0.006	0.008	17688	8958	1
z_beta_over[18]	-0.171	0.733	-1.515	1.253	0.005	0.008	21330	7788	1
z_beta_over[19]	0.554	0.772	-0.871	2.023	0.006	0.009	19639	7355	1
z_beta_over[20]	0.568	0.72	-0.803	1.915	0.005	0.008	21162	7760	1
z_beta_over[21]	-0.038	0.795	-1.566	1.434	0.006	0.008	17072	7569	1
z_beta_over[22]	-0.059	0.721	-1.385	1.349	0.005	0.008	21201	7254	1
z_beta_over[23]	-0.425	0.74	-1.77	0.985	0.005	0.009	20284	7200	1
z_beta_over[24]	-0.815	0.712	-2.124	0.537	0.005	0.009	24327	5757	1
z_beta_over[25]	-1.78	0.727	-3.198	-0.453	0.005	0.009	25163	6833	1
z_beta_over[26]	-1.197	0.731	-2.6	0.139	0.005	0.008	22201	8150	1
z_beta_over[27]	-1.487	0.731	-2.802	-0.061	0.005	0.009	21314	7267	1
z_beta_over[28]	-1.244	0.792	-2.714	0.255	0.006	0.008	17376	7100	1

z_beta_over[29]	0.231	0.743	-1.177	1.609	0.005	0.009	19051	6994	1
z_beta_over[30]	-0.528	0.745	-1.923	0.843	0.005	0.008	20490	7701	1
z_beta_over[31]	0.08	0.739	-1.339	1.394	0.005	0.008	20860	8267	1
z_beta_over[32]	0.689	0.725	-0.718	2.008	0.005	0.009	21383	7193	1
z_beta_over[33]	0.319	0.774	-1.15	1.804	0.006	0.009	19342	7464	1
z_beta_over[34]	0.836	0.715	-0.466	2.212	0.005	0.008	21251	7294	1
z_beta_over[35]	-0.666	0.71	-2.014	0.666	0.005	0.008	20865	7903	1
z_beta_over[36]	-0.025	0.745	-1.41	1.383	0.005	0.008	21281	7441	1
z_beta_over[37]	0.67	0.716	-0.655	2.005	0.005	0.009	21255	6827	1
z_beta_over[38]	-0.588	0.746	-1.996	0.816	0.005	0.009	20454	7050	1
z_beta_over[39]	-0.199	0.706	-1.479	1.187	0.005	0.008	21266	6855	1
z_beta_over[40]	0.289	0.749	-1.087	1.73	0.005	0.009	21405	6102	1
z_beta_over[41]	0.136	0.737	-1.195	1.586	0.005	0.009	20662	6504	1
z_beta_over[42]	-0.382	0.713	-1.813	0.883	0.005	0.009	23594	6562	1
z_beta_over[43]	1.008	0.719	-0.292	2.386	0.005	0.009	24518	6367	1
z_beta_over[44]	0.432	0.732	-0.951	1.78	0.005	0.008	21395	7266	1
z_beta_over[45]	0.858	0.712	-0.456	2.192	0.005	0.009	23324	6857	1
z_beta_over[46]	0.567	0.741	-0.825	1.927	0.005	0.009	22293	6975	1
z_beta_over[47]	0.702	0.719	-0.629	2.062	0.005	0.009	24569	6482	1
z_beta_over[48]	1.208	0.823	-0.365	2.709	0.007	0.009	14743	7883	1
z_beta_over[49]	1.122	0.73	-0.204	2.513	0.005	0.009	26319	6924	1
z_beta_over[50]	0.276	0.72	-1.12	1.603	0.005	0.009	23701	6023	1
z_beta_over[51]	-1.019	0.706	-2.361	0.297	0.005	0.008	23158	7548	1
z_beta_over[52]	-0.34	0.706	-1.662	0.969	0.005	0.008	23648	7962	1
z_beta_over[53]	0.416	0.711	-0.939	1.753	0.005	0.009	24575	6770	1
z_beta_over[54]	-0.403	0.705	-1.715	0.93	0.005	0.009	23483	7337	1
z_beta_over[55]	0.559	0.727	-0.777	1.94	0.005	0.009	22800	6988	1
z_beta_over[56]	0.287	0.715	-1.058	1.649	0.005	0.009	23291	6992	1
z_beta_over[57]	0.1	0.718	-1.257	1.459	0.005	0.008	23597	7459	1
z_beta_over[58]	0.551	0.728	-0.834	1.911	0.005	0.009	20896	7484	1
z_beta_over[59]	-0.208	0.706	-1.522	1.118	0.005	0.008	20019	7210	1

z_beta_over[60]	0.659	0.737	-0.755	2.017	0.005	0.01	24691	6422	1
z_beta_over[61]	0.314	0.719	-1.028	1.671	0.005	0.008	21177	7657	1
z_beta_over[62]	0.261	0.707	-1.05	1.588	0.004	0.008	27379	7159	1
z_beta_over[63]	-0.535	0.707	-1.86	0.786	0.004	0.008	26022	6934	1
z_beta_over[64]	-0.852	0.781	-2.315	0.593	0.006	0.008	16714	8727	1
z_beta_over[65]	-1.314	0.771	-2.712	0.17	0.006	0.008	16420	7848	1
z_beta_over[66]	-0.967	0.753	-2.353	0.484	0.006	0.009	18630	6655	1
z_beta_over[67]	-0.96	0.723	-2.295	0.428	0.005	0.009	20481	6577	1
z_beta_over[68]	0.354	0.817	-1.11	1.914	0.006	0.008	17059	8731	1
z_beta_over[69]	0.824	0.811	-0.707	2.321	0.006	0.008	17434	8374	1
z_beta_over[70]	0.367	0.722	-0.977	1.734	0.005	0.008	21137	7724	1
z_beta_over[71]	0.577	0.733	-0.804	1.952	0.005	0.009	24057	7672	1
z_beta_over[72]	-0.509	0.709	-1.788	0.845	0.005	0.008	24281	7701	1
z_beta_over[73]	1.213	0.709	-0.096	2.578	0.005	0.008	22342	7119	1
z_beta_over[74]	-0.783	0.796	-2.38	0.639	0.006	0.008	17467	8219	1
z_beta_over[75]	0.404	0.713	-0.918	1.763	0.005	0.009	22169	6723	1
z_beta_over[76]	-0.343	0.731	-1.696	1.044	0.005	0.009	25975	7281	1
z_beta_over[77]	0.789	0.747	-0.626	2.168	0.006	0.008	15508	7991	1
z_beta_over[78]	-0.894	0.77	-2.354	0.561	0.005	0.008	20126	7823	1
z_beta_over[79]	-0.299	0.748	-1.718	1.081	0.005	0.008	21154	7662	1
z_beta_over[80]	0.181	0.73	-1.227	1.535	0.005	0.008	20751	6980	1
z_beta_over[81]	0.073	0.731	-1.318	1.44	0.005	0.008	22728	6678	1
z_beta_over[82]	-0.231	0.716	-1.585	1.106	0.005	0.008	21568	7454	1
z_beta_over[83]	0.511	0.734	-0.846	1.906	0.005	0.009	25238	7436	1
z_beta_over[84]	1.01	0.813	-0.506	2.567	0.006	0.009	19141	7776	1
z_beta_over[85]	0.148	0.744	-1.258	1.546	0.005	0.009	18753	6974	1
z_beta_over[86]	1.004	0.726	-0.381	2.334	0.005	0.009	24509	7543	1
z_beta_over[87]	0.716	0.759	-0.714	2.171	0.005	0.008	20707	7625	1
z_beta_over[88]	0.149	0.713	-1.222	1.455	0.005	0.008	22548	7685	1
z_beta_over[89]	1.001	0.725	-0.44	2.268	0.005	0.008	21531	7757	1
z_beta_over[90]	0.452	0.741	-0.96	1.807	0.005	0.008	20166	7301	1

z_beta_over[91]	0.347	0.714	-0.968	1.658	0.005	0.009	24841	6445	1
z_beta_over[92]	-0.389	0.742	-1.782	1.004	0.005	0.009	21021	7161	1
z_beta_over[93]	0.486	0.719	-0.848	1.843	0.005	0.009	25257	7147	1
z_beta_over[94]	0.72	0.797	-0.731	2.273	0.006	0.009	20435	7637	1
z_beta_over[95]	-1.117	0.809	-2.644	0.391	0.006	0.009	20435	7878	1
z_beta_over[96]	-0.198	0.796	-1.723	1.277	0.006	0.008	17572	7467	1
z_beta_over[97]	0.108	0.737	-1.231	1.534	0.005	0.009	18668	7627	1
z_beta_over[98]	0.475	0.738	-0.95	1.796	0.005	0.008	20733	8096	1
z_beta_over[99]	-0.075	0.723	-1.497	1.233	0.005	0.009	25450	6818	1
z_beta_over[100]	0.177	0.731	-1.194	1.52	0.005	0.008	21739	7781	1
z_beta_over[101]	-0.005	0.726	-1.39	1.331	0.005	0.009	23640	6692	1
z_beta_over[102]	-0.066	0.722	-1.399	1.293	0.005	0.008	21877	7485	1
z_beta_over[103]	-0.046	0.708	-1.361	1.308	0.005	0.008	23325	7387	1
z_beta_over[104]	0.174	0.743	-1.248	1.546	0.005	0.009	22625	7269	1
z_beta_over[105]	-0.206	0.721	-1.595	1.123	0.005	0.008	20392	6705	1
z_beta_over[106]	-0.421	0.744	-1.817	0.985	0.005	0.008	21783	7283	1
z_beta_over[107]	-0.475	0.73	-1.836	0.909	0.005	0.007	21368	8257	1
z_beta_over[108]	0.034	0.783	-1.468	1.489	0.006	0.008	17525	8671	1
z_beta_over[109]	0.183	0.752	-1.279	1.545	0.005	0.009	20331	7652	1
z_beta_over[110]	0.543	0.721	-0.757	1.969	0.005	0.009	21795	6736	1
z_beta_over[111]	0.196	0.708	-1.117	1.511	0.005	0.009	22008	7044	1
z_beta_over[112]	0.527	0.742	-0.937	1.874	0.005	0.009	21977	6491	1
z_beta_over[113]	0.41	0.738	-0.903	1.84	0.005	0.008	19868	7446	1
z_beta_over[114]	0.343	0.713	-0.958	1.734	0.005	0.008	20693	7047	1
z_beta_over[115]	-0.275	0.719	-1.644	1.046	0.004	0.009	25721	6891	1

Appendix B

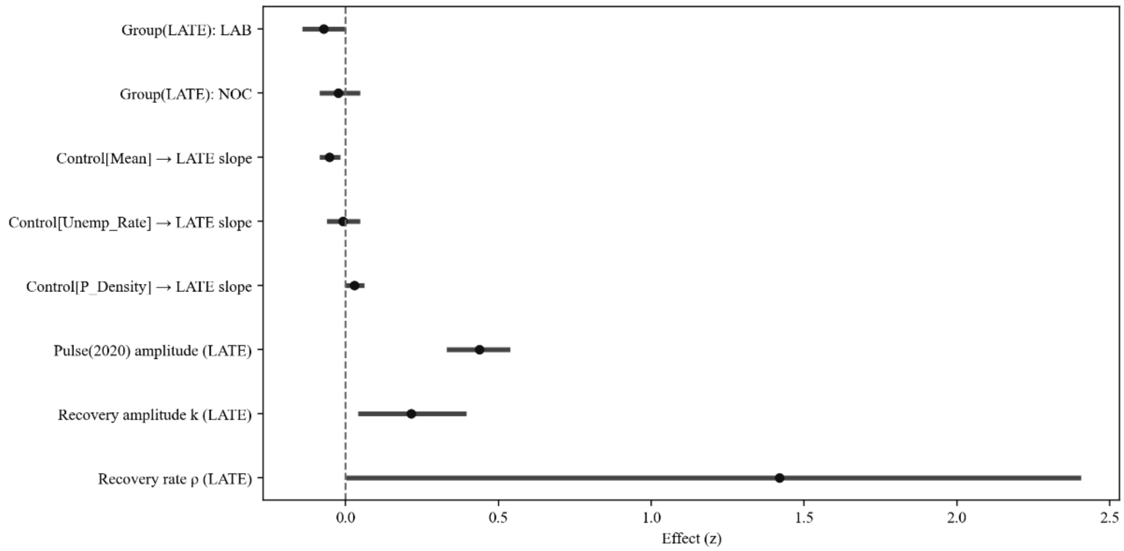

Figure B1. Bayesian Coefficient Forest Plot of Long-Term Party Structure on Per Capita Adult Care Expenditure

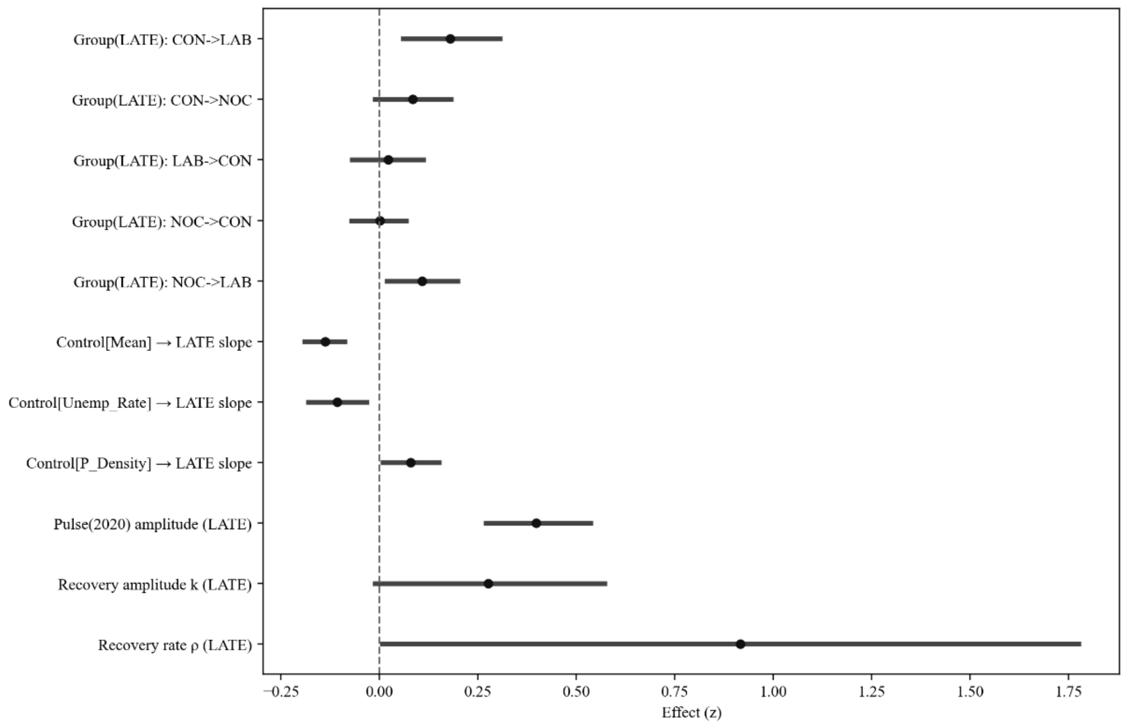

Figure B2. Bayesian Coefficient Forest Plot of Political Alternation on Per Capita Adult Care Expenditure

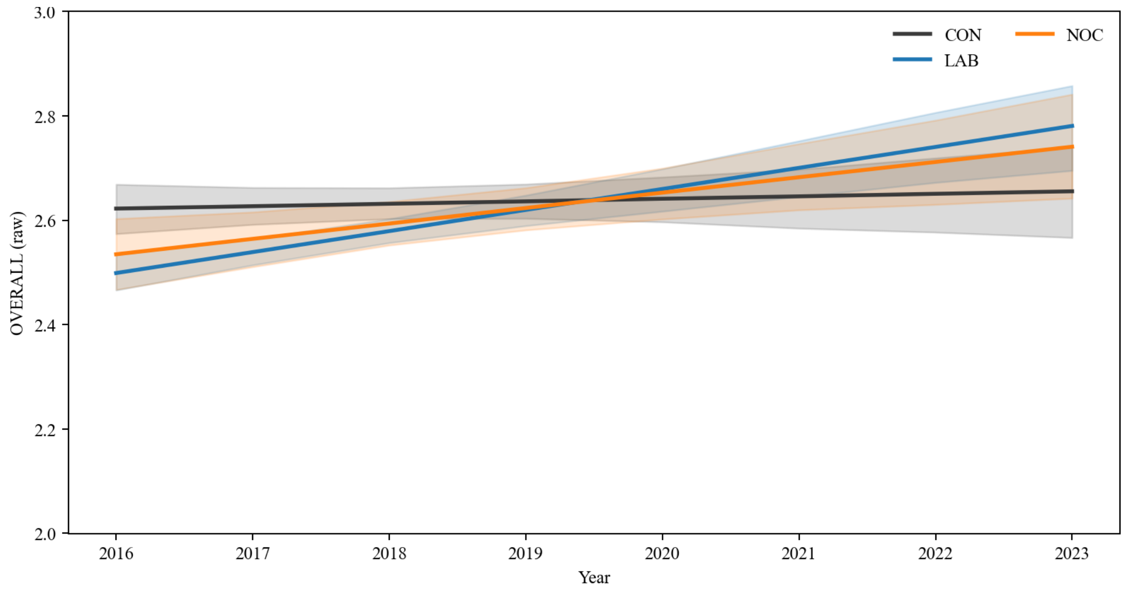

Figure B3. Long-term Party Structure Trends on Care Home Quality (Joint Model)

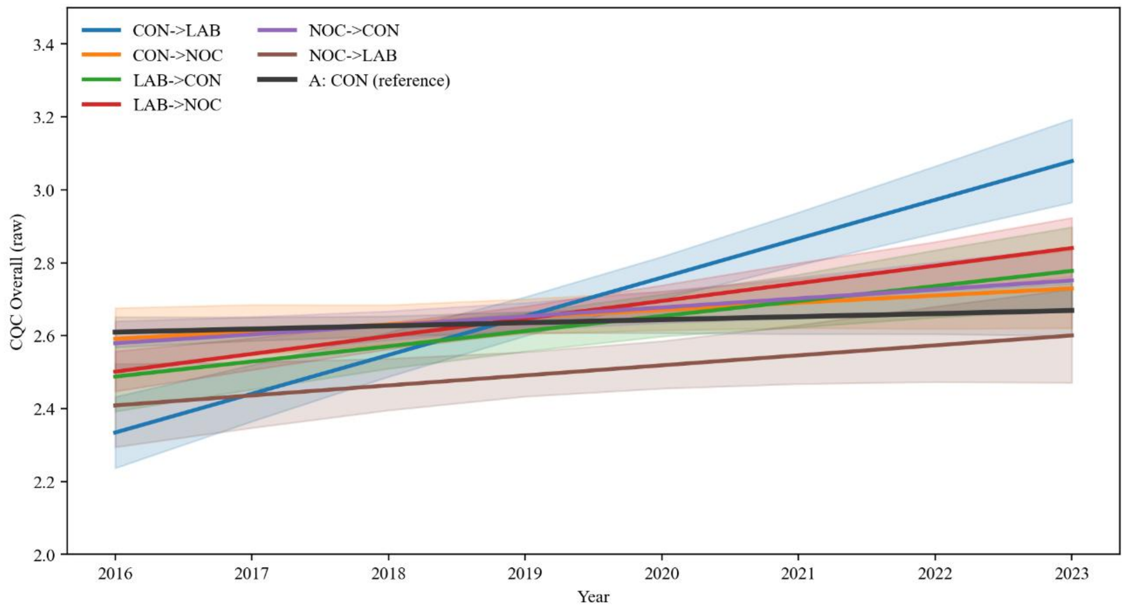

Figure B4. Political Alternation Trends on Care Home Quality (Joint Model)

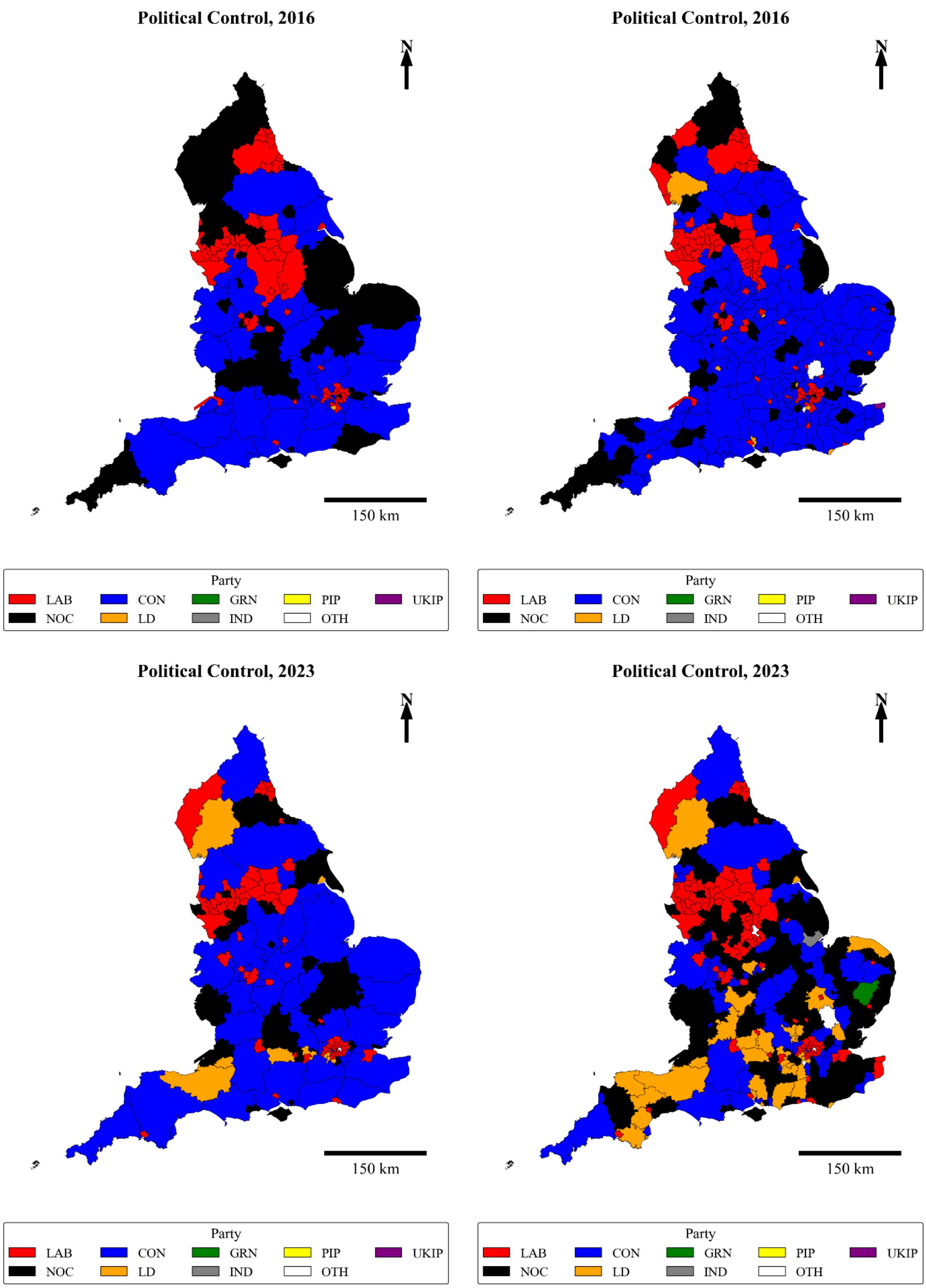

Figure B5. Non-metropolitan District Election Comparison 2016-2023

Appendix C

Table C1. Dataset

ONSCode	Year	Plan_P	Overall	Unemp_Rate	Mean	P_Density	LATE_P18	Covid
E06000001	2016	0	2.22	59.24	24626	942.86	699.68	0
E06000001	2017	0	2.76	80.54	24195	944.91	707.68	0
E06000001	2018	0	2.55	88.68	24514	947.18	744.02	0
E06000001	2019	0	2.62	76.19	23558	952.38	739.03	0
E06000001	2020	2	2.00	110.64	23458	954.13	741.48	1
E06000001	2021	2	3.00	87.38	22731	941.23	723.24	0
E06000001	2022	2	2.91	64.52	21144	954.20	677.08	0
E06000001	2023	2	2.86	61.92	22435	969.67	697.23	0
E06000002	2016	0	2.46	58.75	21072	2573.58	655.04	0
E06000002	2017	0	2.64	57.75	21066	2578.00	654.27	0
E06000002	2018	0	2.33	56.92	21583	2576.27	687.02	0
E06000002	2019	0	2.85	73.02	22779	2584.25	692.23	0
E06000002	2020	2	2.63	126.76	23880	2589.84	781.08	1
E06000002	2021	2	2.43	104.18	21719	2634.73	769.98	0
E06000002	2022	2	2.78	73.49	21501	2723.62	765.38	0
E06000002	2023	2	2.88	68.38	21369	2798.17	782.06	0
E06000003	2016	2	2.73	49.12	22844	533.32	621.87	0
E06000003	2017	2	2.57	45.84	22647	535.68	578.98	0
E06000003	2018	2	2.48	42.57	21921	538.49	665.09	0
E06000003	2019	2	2.71	53.84	21506	540.42	767.24	0
E06000003	2020	2	1.75	91.28	23893	540.72	842.65	1
E06000003	2021	2	2.23	70.53	21543	538.31	797.61	0
E06000003	2022	2	2.25	52.59	21811	540.49	726.64	0
E06000003	2023	2	2.89	50.72	21115	543.50	764.63	0
E06000004	2016	0	2.52	39.84	24669	933.01	569.35	0
E06000004	2017	0	2.55	39.43	24157	936.85	533.94	0
E06000004	2018	0	2.48	42.31	25115	940.31	477.52	0
E06000004	2019	0	2.75	51.80	25094	940.96	509.45	0
E06000004	2020	2	2.60	87.42	25588	941.30	587.86	1
E06000004	2021	2	2.27	68.11	23823	939.44	553.58	0
E06000004	2022	2	2.61	51.29	24048	954.14	465.08	0
E06000004	2023	2	2.57	48.60	23688	965.07	590.49	0
E06000005	2016	0	2.77	40.53	25903	534.98	534.06	0
E06000005	2017	0	2.90	43.72	26068	538.53	509.07	0
E06000005	2018	0	2.79	48.15	25385	539.64	487.16	0
E06000005	2019	0	2.73	56.83	25814	540.84	494.49	0
E06000005	2020	2	2.18	88.68	25591	543.87	555.34	1
E06000005	2021	2	2.20	70.70	25395	548.02	522.86	0
E06000005	2022	2	2.77	52.28	25076	554.05	505.94	0
E06000005	2023	2	2.33	51.94	23810	559.84	550.31	0
E06000006	2016	0	2.31	34.94	25452	1405.01	493.34	0
E06000006	2017	0	2.65	47.88	24643	1412.68	760.26	0
E06000006	2018	0	2.33	53.38	24838	1421.94	778.99	0
E06000006	2019	0	2.82	51.14	24404	1432.77	814.74	0

E06000006	2020	0	3.00	88.92	26936	1436.63	781.23	1
E06000006	2021	0	2.20	67.51	24907	1423.55	732.78	0
E06000006	2022	0	2.00	53.19	24886	1428.32	729.69	0
E06000006	2023	0	2.50	54.56	23843	1434.98	748.93	0
E06000007	2016	0	2.70	20.17	28564	1144.89	528.80	0
E06000007	2017	0	2.32	26.05	27676	1149.80	566.95	0
E06000007	2018	0	2.52	32.43	28153	1148.94	568.25	0
E06000007	2019	0	2.80	31.49	28453	1151.50	566.81	0
E06000007	2020	0	2.00	64.78	26868	1148.11	638.85	1
E06000007	2021	0	2.47	48.85	27435	1158.15	665.66	0
E06000007	2022	0	2.71	33.59	27250	1161.27	645.89	0
E06000007	2023	0	2.45	31.00	27921	1164.51	705.57	0
E06000008	2016	0	2.75	35.50	21501	1073.18	614.03	0
E06000008	2017	0	2.68	36.26	21438	1085.75	612.30	0
E06000008	2018	0	2.65	49.28	20942	1086.99	646.43	0
E06000008	2019	0	2.28	59.86	21998	1092.50	648.30	0
E06000008	2020	0	2.29	102.07	21976	1094.93	683.97	1
E06000008	2021	0	2.31	83.46	20522	1130.64	669.79	0
E06000008	2022	0	2.47	66.00	20469	1137.21	641.15	0
E06000008	2023	0	1.83	65.39	21565	1149.51	682.38	0
E06000009	2016	0	2.55	50.32	20079	3225.78	613.87	0
E06000009	2017	0	2.80	55.11	19635	3241.43	653.90	0
E06000009	2018	0	2.97	56.82	19982	3228.33	687.89	0
E06000009	2019	0	2.92	78.79	20540	3231.60	698.84	0
E06000009	2020	0	2.13	150.84	20855	3206.92	854.91	1
E06000009	2021	0	2.23	114.15	19999	3266.55	815.00	0
E06000009	2022	0	2.75	85.56	19573	3282.63	748.40	0
E06000009	2023	0	2.61	77.58	19675	3307.19	769.35	0
E06000012	2016	2	2.62	43.59	23043	781.83	430.46	0
E06000012	2017	2	2.60	43.46	22046	785.18	436.40	0
E06000012	2018	2	2.63	53.99	22754	785.15	439.64	0
E06000012	2019	2	2.81	54.42	23483	787.33	446.25	0
E06000012	2020	1	2.00	89.81	24912	786.35	494.41	1
E06000012	2021	1	2.50	68.90	24531	775.66	486.51	0
E06000012	2022	1	2.50	53.27	23716	778.36	439.72	0
E06000012	2023	1	2.50	53.59	25290	781.36	489.02	0
E06000013	2016	1	2.58	28.97	24362	195.03	444.83	0
E06000013	2017	1	2.79	27.59	26819	195.61	411.42	0
E06000013	2018	1	2.70	38.59	23978	196.42	399.94	0
E06000013	2019	1	2.79	41.30	24308	196.75	419.39	0
E06000013	2020	1	2.09	73.95	24439	196.55	456.03	1
E06000013	2021	1	2.06	58.12	25558	193.36	494.01	0
E06000013	2022	1	2.50	44.13	24810	193.52	484.04	0
E06000013	2023	1	2.63	43.59	24792	193.51	515.56	0
E06000014	2016	2	2.41	8.28	26399	766.01	433.80	0
E06000014	2017	2	2.65	10.77	25533	765.26	432.69	0
E06000014	2018	2	2.60	15.74	24326	771.62	472.14	0
E06000014	2019	2	2.63	15.88	28905	774.29	518.13	0
E06000014	2020	2	2.57	43.05	29107	775.74	552.74	1

E06000014	2021	2	2.00	32.78	27580	741.41	575.68	0
E06000014	2022	2	2.64	22.59	26833	750.40	518.28	0
E06000014	2023	2	2.44	20.91	25933	760.16	509.55	0
E06000015	2016	0	2.65	19.10	26962	3283.73	480.43	0
E06000015	2017	0	2.42	21.76	26650	3293.99	465.93	0
E06000015	2018	0	2.51	28.40	26914	3295.79	467.11	0
E06000015	2019	2	2.58	41.21	27166	3297.43	469.02	0
E06000015	2020	2	2.38	92.01	27983	3291.17	516.79	1
E06000015	2021	2	2.25	76.02	26185	3346.56	485.78	0
E06000015	2022	2	2.54	58.04	24559	3378.40	448.49	0
E06000015	2023	2	2.69	61.14	24894	3415.27	471.86	0
E06000016	2016	0	2.74	21.89	21323	4749.57	553.01	0
E06000016	2017	0	2.65	21.70	20630	4820.43	507.47	0
E06000016	2018	0	2.60	26.17	19560	4843.30	527.15	0
E06000016	2019	0	2.73	38.88	21635	4829.75	544.59	0
E06000016	2020	0	2.00	90.15	22061	4827.19	638.70	1
E06000016	2021	0	2.23	76.86	21588	4990.56	660.75	0
E06000016	2022	0	2.19	58.83	21520	5078.87	643.17	0
E06000016	2023	0	2.39	61.69	21125	5178.35	654.43	0
E06000017	2016	1	3.00	6.80	28513	98.05	493.83	0
E06000017	2017	1	2.83	7.47	28550	100.25	411.84	0
E06000017	2018	1	2.67	12.04	23340	100.82	533.36	0
E06000017	2019	1	3.00	14.67	29899	101.40	505.28	0
E06000017	2020	1	2.00	41.25	26930	102.80	570.71	1
E06000017	2021	1	2.67	28.89	28864	105.09	541.38	0
E06000017	2022	1	2.20	18.47	30500	104.70	532.72	0
E06000017	2023	1	2.00	21.59	30471	103.22	593.92	0
E06000018	2016	0	2.43	37.65	22320	4359.56	553.24	0
E06000018	2017	0	2.49	38.88	22558	4412.19	563.99	0
E06000018	2018	0	2.54	37.77	21602	4437.12	578.57	0
E06000018	2019	0	2.77	50.25	22081	4461.66	593.74	0
E06000018	2020	0	1.91	92.93	22691	4517.92	588.96	1
E06000018	2021	0	1.86	86.23	21002	4282.95	595.46	0
E06000018	2022	0	2.11	65.91	20340	4388.26	585.59	0
E06000018	2023	0	2.00	68.02	19680	4412.38	612.91	0
E06000019	2016	1	2.78	13.00	25097	86.85	470.60	0
E06000019	2017	1	2.28	12.74	21849	87.65	507.14	0
E06000019	2018	1	2.74	15.58	23538	88.13	488.76	0
E06000019	2019	1	2.59	21.45	23921	88.45	477.87	0
E06000019	2020	2	2.50	52.15	24036	88.83	541.71	1
E06000019	2021	2	2.65	40.44	23336	86.05	566.55	0
E06000019	2022	2	2.29	29.82	22709	86.57	557.46	0
E06000019	2023	2	2.73	30.01	22977	87.12	595.65	0
E06000020	2016	2	2.67	19.77	24097	595.82	431.75	0
E06000020	2017	2	2.19	22.08	22787	605.44	424.16	0
E06000020	2018	2	2.68	22.76	22506	612.44	443.70	0
E06000020	2019	2	2.70	34.58	23953	619.52	471.30	0
E06000020	2020	0	2.38	76.34	24233	624.57	513.23	1
E06000020	2021	0	1.78	57.75	22667	640.14	507.17	0

E06000020	2022	0	2.33	43.10	23703	651.02	578.24	0
E06000020	2023	0	1.71	42.34	23689	661.15	608.74	0
E06000021	2016	2	2.35	26.23	22419	2709.79	570.89	0
E06000021	2017	2	2.37	29.97	22335	2732.82	562.88	0
E06000021	2018	2	2.64	37.02	23115	2737.69	569.24	0
E06000021	2019	2	2.57	52.28	23391	2743.49	545.44	0
E06000021	2020	2	2.20	99.02	24372	2746.13	616.86	1
E06000021	2021	2	2.31	84.14	22843	2761.28	614.59	0
E06000021	2022	2	2.44	66.76	21793	2782.37	633.22	0
E06000021	2023	2	2.30	68.56	21648	2816.44	654.77	0
E06000023	2016	2	2.49	20.77	26253	1930.12	556.15	0
E06000023	2017	0	2.75	20.78	26136	1951.54	568.27	0
E06000023	2018	0	2.75	22.22	26476	1969.18	568.20	0
E06000023	2019	0	2.88	31.30	27403	1968.26	571.56	0
E06000023	2020	0	2.50	75.64	27646	1978.69	632.94	1
E06000023	2021	2	2.69	57.52	27314	2000.99	611.25	0
E06000023	2022	2	2.81	40.43	26917	2032.93	606.30	0
E06000023	2023	2	2.86	41.19	27027	2046.44	622.59	0
E06000024	2016	1	2.50	12.43	28851	541.64	511.78	0
E06000024	2017	1	2.53	15.00	27605	544.59	514.37	0
E06000024	2018	1	2.50	22.45	28205	547.37	518.55	0
E06000024	2019	1	2.42	25.99	27409	550.42	539.25	0
E06000024	2020	2	2.33	58.40	31066	552.09	606.66	1
E06000024	2021	2	2.58	41.65	26166	556.77	581.15	0
E06000024	2022	2	2.46	28.64	27137	561.29	555.31	0
E06000024	2023	2	2.50	29.28	27135	566.31	599.72	0
E06000025	2016	1	2.66	10.88	27106	517.33	470.42	0
E06000025	2017	1	2.70	11.04	26162	519.95	512.98	0
E06000025	2018	1	2.98	11.28	26239	526.69	495.17	0
E06000025	2019	1	3.05	16.41	25904	531.47	512.84	0
E06000025	2020	1	3.00	46.27	28457	536.21	592.81	1
E06000025	2021	1	2.88	32.36	26248	541.65	574.08	0
E06000025	2022	1	2.75	20.97	26470	550.17	559.69	0
E06000025	2023	1	2.54	21.35	26946	557.90	587.46	0
E06000026	2016	2	2.36	26.77	21880	3131.71	455.81	0
E06000026	2017	2	2.60	26.34	22769	3118.32	457.00	0
E06000026	2018	2	2.60	36.26	23605	3118.68	477.71	0
E06000026	2019	0	2.73	40.97	22592	3106.56	492.15	0
E06000026	2020	0	2.63	77.30	22669	3115.32	542.52	1
E06000026	2021	2	2.19	57.43	21268	3137.70	552.37	0
E06000026	2022	2	2.71	40.24	21140	3165.39	507.28	0
E06000026	2023	2	2.27	40.88	22633	3185.38	525.19	0
E06000027	2016	1	2.54	24.31	20258	1120.67	395.04	0
E06000027	2017	1	2.52	25.58	20753	1132.09	407.26	0
E06000027	2018	1	2.57	27.59	20481	1136.55	418.86	0
E06000027	2019	1	2.49	39.41	21597	1140.60	398.73	0
E06000027	2020	2	2.13	93.45	21357	1140.22	508.95	1
E06000027	2021	2	2.38	60.11	22302	1167.38	377.72	0
E06000027	2022	2	2.39	40.22	21255	1167.07	687.28	0

E06000027	2023	2	2.57	39.53	21967	1167.72	776.12	0
E06000031	2016	2	2.67	17.40	24775	573.89	447.33	0
E06000031	2017	1	2.85	18.66	24605	579.18	465.93	0
E06000031	2018	1	2.90	33.79	24748	585.38	436.56	0
E06000031	2019	1	2.89	42.55	24588	588.92	441.94	0
E06000031	2020	2	3.00	95.23	24702	589.99	503.58	1
E06000031	2021	2	2.57	78.21	25445	629.95	475.26	0
E06000031	2022	2	2.67	60.38	24510	633.76	458.71	0
E06000031	2023	2	2.50	66.27	25753	639.09	506.07	0
E06000032	2016	0	2.60	22.40	23959	5000.66	419.00	0
E06000032	2017	0	2.44	23.79	25670	4951.46	484.81	0
E06000032	2018	0	2.32	24.31	23425	4938.79	483.65	0
E06000032	2019	0	2.43	37.42	26160	4914.41	487.29	0
E06000032	2020	0	1.83	104.94	25942	4925.39	513.10	1
E06000032	2021	0	2.60	89.83	23829	5186.00	478.79	0
E06000032	2022	0	1.75	66.69	23427	5243.02	465.19	0
E06000032	2023	0	2.86	70.91	23244	5329.05	488.06	0
E06000033	2016	2	2.53	24.72	30101	2647.96	473.47	0
E06000033	2017	2	2.58	28.10	27412	2677.55	467.91	0
E06000033	2018	2	2.58	41.32	29569	2687.19	466.86	0
E06000033	2019	1	2.47	46.03	30602	2698.41	486.86	0
E06000033	2020	2	2.00	97.34	30522	2693.23	532.55	1
E06000033	2021	2	2.31	76.18	30326	2657.30	500.40	0
E06000033	2022	2	2.27	54.57	29341	2661.46	512.81	0
E06000033	2023	2	2.31	53.75	28848	2681.64	539.89	0
E06000034	2016	2	3.00	24.00	26963	906.18	510.49	0
E06000034	2017	2	2.89	22.99	27937	924.46	512.17	0
E06000034	2018	2	2.82	30.27	26538	936.02	438.71	0
E06000034	2019	2	2.79	37.55	28491	945.87	573.08	0
E06000034	2020	2	2.00	85.94	28937	951.85	520.61	1
E06000034	2021	1	3.33	66.47	29891	953.75	500.77	0
E06000034	2022	1	2.67	47.03	28496	958.56	551.52	0
E06000034	2023	1	2.40	51.24	29186	966.19	488.45	0
E06000035	2016	1	2.64	24.06	27587	1035.23	398.94	0
E06000035	2017	1	2.67	23.91	26995	1031.79	413.95	0
E06000035	2018	1	2.59	27.31	28131	1032.67	410.44	0
E06000035	2019	1	2.38	38.50	28288	1035.28	421.44	0
E06000035	2020	1	2.00	84.03	29835	1038.00	481.82	1
E06000035	2021	1	2.35	69.11	27086	1040.55	456.47	0
E06000035	2022	1	2.20	48.00	26726	1051.02	457.04	0
E06000035	2023	1	2.20	47.80	26863	1066.42	477.56	0
E06000036	2016	1	2.94	9.77	32625	1091.99	481.00	0
E06000036	2017	1	3.00	9.48	32165	1100.50	477.96	0
E06000036	2018	1	2.44	11.70	30979	1112.37	443.71	0
E06000036	2019	1	2.50	17.97	33315	1120.35	478.17	0
E06000036	2020	1	2.00	53.67	32121	1135.13	499.53	1
E06000036	2021	1	2.50	37.63	32748	1144.35	498.03	0
E06000036	2022	1	1.67	25.00	31037	1161.32	492.67	0
E06000036	2023	1	2.00	25.19	33933	1173.50	545.01	0

E06000037	2016	1	2.82	7.85	35191	222.73	506.79	0
E06000037	2017	1	2.90	8.23	32792	225.05	509.95	0
E06000037	2018	1	2.76	13.67	33708	225.13	559.58	0
E06000037	2019	1	2.82	17.91	35575	225.02	582.83	0
E06000037	2020	1	2.33	49.63	32104	225.04	650.91	1
E06000037	2021	1	2.80	36.83	35606	229.87	640.37	0
E06000037	2022	1	2.00	26.46	29978	230.62	640.06	0
E06000037	2023	1	2.00	25.66	31357	232.00	656.85	0
E06000038	2016	0	2.91	17.81	29865	4026.59	466.92	0
E06000038	2017	0	2.85	19.55	29748	4036.71	438.63	0
E06000038	2018	0	2.63	26.04	30632	4039.88	416.74	0
E06000038	2019	0	2.50	33.17	30500	4004.65	434.44	0
E06000038	2020	0	2.00	78.86	31512	3968.93	494.68	1
E06000038	2021	0	2.75	58.96	33247	4286.60	463.91	0
E06000038	2022	0	1.75	44.82	29708	4350.27	400.10	0
E06000038	2023	0	2.20	47.42	29499	4411.49	467.72	0
E06000041	2016	1	2.81	7.43	39644	904.52	447.24	0
E06000041	2017	1	2.97	7.03	37953	921.85	453.32	0
E06000041	2018	1	2.92	9.38	39565	938.61	441.19	0
E06000041	2019	1	2.59	12.53	36701	956.16	466.08	0
E06000041	2020	1	3.00	40.70	36614	971.95	531.01	1
E06000041	2021	1	2.27	27.99	37910	995.55	492.87	0
E06000041	2022	1	2.17	19.12	39584	1013.49	476.17	0
E06000041	2023	2	2.00	21.07	34735	1027.36	482.14	0
E06000042	2016	2	2.57	18.01	29831	856.95	393.03	0
E06000042	2017	2	2.90	19.39	30111	866.81	390.84	0
E06000042	2018	2	2.63	19.54	29569	870.33	395.65	0
E06000042	2019	2	2.81	28.35	30335	873.08	395.42	0
E06000042	2020	2	2.50	76.09	31107	875.50	493.17	1
E06000042	2021	2	2.11	58.05	30758	933.82	433.94	0
E06000042	2022	2	2.67	42.98	30555	947.80	422.33	0
E06000042	2023	2	2.78	45.02	29249	966.38	464.11	0
E06000043	2016	2	2.73	17.46	29881	3387.68	491.24	0
E06000043	2017	2	2.59	18.51	28829	3375.10	490.16	0
E06000043	2018	2	2.74	26.97	29467	3401.33	515.97	0
E06000043	2019	2	2.64	31.94	28686	3406.71	538.73	0
E06000043	2020	2	2.22	81.56	28859	3416.70	582.79	1
E06000043	2021	2	2.54	68.42	27476	3236.30	612.54	0
E06000043	2022	2	2.26	49.64	27027	3260.14	603.43	0
E06000043	2023	2	2.44	48.98	26305	3275.41	622.01	0
E06000044	2016	2	2.33	17.43	23971	3571.94	406.32	0
E06000044	2017	2	1.88	19.98	23556	3570.04	424.38	0
E06000044	2018	2	2.39	22.09	23475	3576.94	441.78	0
E06000044	2019	2	2.30	33.76	24021	3573.66	463.15	0
E06000044	2020	2	2.17	85.28	25706	3569.92	497.43	1
E06000044	2021	2	2.00	72.79	24512	3439.16	527.92	0
E06000044	2022	2	1.69	55.57	25056	3474.87	506.96	0
E06000044	2023	2	2.50	56.10	24169	3496.22	498.34	0
E06000045	2016	0	2.61	16.68	23240	4509.61	438.69	0

E06000045	2017	0	2.71	24.51	23515	4475.63	445.34	0
E06000045	2018	0	2.92	34.34	22605	4483.38	501.11	0
E06000045	2019	0	2.41	37.90	24204	4478.48	499.66	0
E06000045	2020	0	2.75	84.05	26444	4484.73	561.69	1
E06000045	2021	1	2.83	69.65	26573	4385.13	606.39	0
E06000045	2022	1	2.33	50.37	25991	4471.94	573.39	0
E06000045	2023	0	2.64	49.58	23774	4542.15	546.79	0
E06000046	2016	2	2.57	22.45	23311	353.98	647.18	0
E06000046	2017	2	2.30	23.20	22738	356.99	625.50	0
E06000046	2018	1	2.58	25.21	22092	360.30	608.71	0
E06000046	2019	1	2.85	34.85	22463	360.90	611.79	0
E06000046	2020	1	2.25	82.02	24365	362.24	664.20	1
E06000046	2021	2	2.64	61.19	23169	358.66	690.90	0
E06000046	2022	2	2.70	44.67	21799	358.38	689.66	0
E06000046	2023	2	2.00	45.78	20767	358.70	691.99	0
E06000047	2016	0	2.63	29.07	24672	233.86	528.32	0
E06000047	2017	0	2.81	32.25	24826	234.54	520.10	0
E06000047	2018	0	2.53	43.78	25126	236.03	515.68	0
E06000047	2019	0	2.74	48.19	25616	237.43	525.81	0
E06000047	2020	0	2.44	78.57	25256	238.80	565.36	1
E06000047	2021	2	2.53	62.78	24952	233.51	564.85	0
E06000047	2022	2	2.94	42.75	24440	236.36	546.58	0
E06000047	2023	2	2.81	39.85	23418	238.37	576.89	0
E06000049	2016	1	2.60	14.06	30127	322.97	466.03	0
E06000049	2017	1	2.29	15.67	30489	324.81	450.64	0
E06000049	2018	1	2.50	20.35	29308	326.48	481.56	0
E06000049	2019	1	2.62	24.02	31412	329.36	509.12	0
E06000049	2020	2	2.50	57.71	32394	331.52	552.23	1
E06000049	2021	2	2.23	40.13	28492	343.40	544.57	0
E06000049	2022	2	2.46	28.54	29704	348.60	527.25	0
E06000049	2023	2	2.38	28.23	29694	353.64	544.43	0
E06000050	2016	0	2.10	17.27	26912	356.65	441.85	0
E06000050	2017	0	2.31	19.49	27793	359.10	448.76	0
E06000050	2018	0	2.59	28.85	27531	361.77	467.52	0
E06000050	2019	0	2.60	31.44	28367	364.50	510.11	0
E06000050	2020	2	1.89	66.04	28343	365.30	554.68	1
E06000050	2021	2	2.25	44.82	28549	380.04	542.51	0
E06000050	2022	2	2.40	31.87	28150	384.40	545.75	0
E06000050	2023	2	2.46	30.62	26825	387.87	590.29	0
E06000051	2016	1	2.81	12.76	26110	98.01	447.67	0
E06000051	2017	1	2.65	13.75	25485	99.29	483.55	0
E06000051	2018	1	2.79	18.06	24958	100.17	493.07	0
E06000051	2019	1	2.74	24.46	26152	101.07	507.85	0
E06000051	2020	1	1.86	56.49	25974	101.78	597.42	1
E06000051	2021	1	2.00	42.00	25176	101.56	607.34	0
E06000051	2022	1	2.08	29.99	25773	102.42	594.98	0
E06000051	2023	1	2.71	28.62	25635	102.98	609.01	0
E06000052	2016	2	2.55	17.14	21804	153.23	487.82	0
E06000052	2017	2	2.54	16.47	21891	155.35	516.32	0

E06000052	2018	2	2.69	21.87	21592	156.63	532.38	0
E06000052	2019	2	2.64	29.41	21548	157.69	553.07	0
E06000052	2020	2	2.27	72.57	21875	158.72	650.45	1
E06000052	2021	1	2.36	49.94	21414	158.37	612.55	0
E06000052	2022	1	2.53	33.86	22571	159.34	597.56	0
E06000052	2023	1	2.37	34.69	22842	160.12	623.04	0
E06000054	2016	1	2.45	11.01	26059	150.03	435.77	0
E06000054	2017	1	2.46	13.00	26670	152.38	458.63	0
E06000054	2018	1	2.41	18.68	27814	153.00	486.88	0
E06000054	2019	1	2.50	20.61	27488	153.60	491.70	0
E06000054	2020	1	2.52	50.26	27386	154.84	524.33	1
E06000054	2021	1	2.35	36.86	27274	157.71	524.36	0
E06000054	2022	1	2.43	25.30	26637	158.54	530.47	0
E06000054	2023	1	2.46	25.44	26975	159.11	566.74	0
E06000055	2016	2	2.46	24.77	28111	354.22	503.32	0
E06000055	2017	2	2.92	30.22	28382	356.65	530.25	0
E06000055	2018	2	2.79	39.07	27305	360.24	544.11	0
E06000055	2019	2	2.53	41.25	28581	363.75	537.48	0
E06000055	2020	2	2.25	83.44	30769	366.67	567.24	1
E06000055	2021	2	2.31	62.75	28228	389.92	535.48	0
E06000055	2022	2	2.25	47.01	28075	393.58	533.41	0
E06000055	2023	2	2.45	50.02	26593	398.58	576.57	0
E06000056	2016	1	2.58	9.41	31025	389.76	443.58	0
E06000056	2017	1	2.48	10.32	30469	391.29	432.04	0
E06000056	2018	1	2.46	11.62	30397	396.28	452.10	0
E06000056	2019	1	2.46	16.67	31069	403.33	466.70	0
E06000056	2020	1	2.00	50.78	32024	410.94	460.30	1
E06000056	2021	1	2.38	38.31	30064	412.96	476.16	0
E06000056	2022	1	2.08	27.07	30766	421.73	461.61	0
E06000056	2023	1	2.22	27.03	30040	430.79	489.88	0
E06000057	2016	2	2.55	30.19	25033	62.23	469.08	0
E06000057	2017	2	2.64	31.14	25303	62.82	480.58	0
E06000057	2018	2	2.56	32.32	23998	63.07	515.09	0
E06000057	2019	2	2.89	40.65	24326	63.50	542.27	0
E06000057	2020	2	2.39	73.67	25940	63.77	530.73	1
E06000057	2021	1	2.52	56.85	23700	63.32	609.17	0
E06000057	2022	1	2.63	41.12	24681	63.86	567.30	0
E06000057	2023	1	2.55	39.00	25109	64.40	618.33	0
E08000001	2016	0	2.63	38.38	23323	2025.26	467.54	0
E08000001	2017	0	2.63	40.60	23034	2037.40	487.13	0
E08000001	2018	0	2.44	41.66	23114	2041.40	533.97	0
E08000001	2019	0	2.81	54.62	24344	2056.98	551.01	0
E08000001	2020	2	2.00	105.34	24367	2061.98	584.00	1
E08000001	2021	2	2.75	86.84	23407	2117.72	571.63	0
E08000001	2022	2	2.40	69.48	24797	2139.99	560.92	0
E08000001	2023	2	2.79	68.55	23551	2163.23	586.84	0
E08000002	2016	0	2.87	30.60	26256	1896.93	600.75	0
E08000002	2017	0	2.52	30.82	25986	1906.57	545.26	0
E08000002	2018	0	2.77	33.65	25588	1911.40	628.97	0

E08000002	2019	0	2.63	42.92	26215	1920.27	662.42	0
E08000002	2020	0	1.80	87.50	26906	1917.43	621.59	1
E08000002	2021	0	2.11	69.40	26150	1948.75	611.41	0
E08000002	2022	0	2.48	54.52	26095	1956.46	592.41	0
E08000002	2023	0	2.73	52.85	25089	1965.40	584.68	0
E08000003	2016	0	2.24	33.36	23270	4680.25	438.66	0
E08000003	2017	0	2.08	32.14	22756	4716.90	437.07	0
E08000003	2018	0	2.32	42.97	23810	4735.28	447.13	0
E08000003	2019	0	2.61	52.82	23863	4780.52	484.81	0
E08000003	2020	0	2.31	108.31	24706	4805.44	511.66	1
E08000003	2021	0	2.48	95.54	23817	4754.52	521.83	0
E08000003	2022	0	2.79	72.05	23547	4900.87	525.69	0
E08000003	2023	0	2.40	72.57	24298	5014.27	571.53	0
E08000004	2016	0	2.30	36.83	22842	1634.93	472.34	0
E08000004	2017	0	2.57	42.83	22976	1642.20	469.48	0
E08000004	2018	0	2.71	55.74	22265	1655.30	520.74	0
E08000004	2019	0	2.50	59.96	22211	1665.74	556.58	0
E08000004	2020	0	2.14	117.70	23501	1669.38	593.23	1
E08000004	2021	0	2.38	103.20	24669	1700.60	583.11	0
E08000004	2022	0	2.30	81.13	22479	1714.10	603.84	0
E08000004	2023	0	3.00	80.98	22782	1729.11	610.98	0
E08000006	2016	0	2.24	34.31	24038	2558.98	477.58	0
E08000006	2017	0	2.12	34.20	23764	2585.79	445.20	0
E08000006	2018	0	2.68	37.32	23498	2617.44	471.29	0
E08000006	2019	0	2.74	48.60	23951	2662.98	480.18	0
E08000006	2020	0	2.43	100.66	27023	2702.72	546.05	1
E08000006	2021	0	2.86	80.25	23753	2785.72	498.39	0
E08000006	2022	0	2.36	61.00	24961	2869.08	477.09	0
E08000006	2023	0	2.50	60.90	24520	2922.98	502.96	0
E08000007	2016	2	2.08	23.51	27747	2305.27	540.56	0
E08000007	2017	2	2.60	23.85	27961	2309.14	551.73	0
E08000007	2018	2	2.61	24.03	28653	2314.93	587.66	0
E08000007	2019	2	2.78	32.48	29421	2328.01	598.92	0
E08000007	2020	2	1.80	73.81	29613	2334.15	616.53	1
E08000007	2021	2	2.44	57.01	28689	2342.45	642.92	0
E08000007	2022	2	2.25	42.45	27812	2357.90	611.18	0
E08000007	2023	2	2.64	39.13	27365	2376.35	620.26	0
E08000008	2016	0	2.00	33.25	21883	2163.64	468.61	0
E08000008	2017	0	2.36	33.99	22387	2172.66	518.79	0
E08000008	2018	0	2.32	42.65	22237	2183.11	488.67	0
E08000008	2019	0	2.79	49.53	22737	2195.67	497.28	0
E08000008	2020	0	2.00	98.04	22886	2201.72	555.45	1
E08000008	2021	0	2.13	75.62	23123	2241.29	547.57	0
E08000008	2022	0	2.44	58.15	23852	2256.87	557.78	0
E08000008	2023	0	2.50	58.00	23292	2274.95	579.78	0
E08000009	2016	1	2.18	20.56	33649	2212.96	413.56	0
E08000009	2017	1	2.15	21.42	32040	2220.69	422.69	0
E08000009	2018	1	2.53	28.14	34281	2228.96	438.78	0
E08000009	2019	2	2.43	30.18	33365	2238.24	483.72	0

E08000009	2020	0	2.25	66.04	32975	2240.37	557.79	1
E08000009	2021	0	2.89	51.03	31530	2221.20	574.78	0
E08000009	2022	0	2.93	38.51	29856	2231.62	522.20	0
E08000009	2023	0	3.07	38.33	29316	2239.49	555.39	0
E08000010	2016	0	2.55	32.44	24027	1716.84	446.43	0
E08000010	2017	0	2.59	32.95	23720	1725.29	455.86	0
E08000010	2018	0	2.77	39.85	24148	1732.93	475.01	0
E08000010	2019	0	2.74	46.28	24145	1746.61	495.14	0
E08000010	2020	0	2.33	84.08	25259	1757.51	548.89	1
E08000010	2021	0	2.79	64.70	25750	1752.44	547.69	0
E08000010	2022	0	2.85	47.77	23631	1775.06	517.41	0
E08000010	2023	0	3.00	45.55	23628	1802.60	543.79	0
E08000011	2016	0	2.36	41.32	23822	1710.00	710.48	0
E08000011	2017	0	2.32	43.32	23541	1717.46	719.03	0
E08000011	2018	0	2.38	48.43	23680	1729.14	733.04	0
E08000011	2019	0	2.42	57.50	23993	1744.07	680.69	0
E08000011	2020	0	1.83	106.09	24747	1762.45	667.92	1
E08000011	2021	0	1.83	81.93	24492	1791.61	725.09	0
E08000011	2022	0	2.75	59.97	24039	1816.27	713.37	0
E08000011	2023	0	2.43	56.63	23332	1840.96	757.94	0
E08000012	2016	0	2.29	41.30	24509	3628.88	537.24	0
E08000012	2017	0	2.24	42.99	24468	3681.08	535.20	0
E08000012	2018	0	2.43	43.25	24110	3705.54	558.67	0
E08000012	2019	0	2.60	54.60	24742	3729.71	601.83	0
E08000012	2020	0	2.06	106.60	26495	3747.92	675.18	1
E08000012	2021	0	1.91	93.60	25277	3628.21	683.90	0
E08000012	2022	0	2.03	69.66	24250	3713.29	651.81	0
E08000012	2023	0	1.90	68.42	25208	3771.42	668.10	0
E08000013	2016	0	2.48	35.36	24696	1308.72	561.64	0
E08000013	2017	0	2.67	34.69	23929	1315.14	565.39	0
E08000013	2018	0	2.78	39.40	24412	1320.41	546.65	0
E08000013	2019	0	2.18	47.48	25219	1324.34	553.11	0
E08000013	2020	0	2.67	85.58	25653	1328.08	587.02	1
E08000013	2021	0	2.33	66.35	26453	1344.92	611.88	0
E08000013	2022	0	2.50	50.40	26606	1354.72	582.29	0
E08000013	2023	0	2.20	48.36	25342	1363.93	622.44	0
E08000014	2016	0	2.39	31.68	23834	1339.30	557.98	0
E08000014	2017	0	2.47	33.68	24757	1340.90	555.40	0
E08000014	2018	0	2.64	44.40	24431	1344.84	574.39	0
E08000014	2019	0	2.79	47.57	25080	1363.27	591.14	0
E08000014	2020	0	2.29	88.73	24576	1360.75	634.73	1
E08000014	2021	0	2.48	67.16	24181	1381.45	602.92	0
E08000014	2022	0	2.38	48.62	23407	1388.10	582.52	0
E08000014	2023	0	2.65	46.28	23920	1396.54	629.00	0
E08000015	2016	0	2.43	26.41	26055	1252.88	519.75	0
E08000015	2017	0	2.32	28.15	26177	1258.96	533.20	0
E08000015	2018	0	2.30	40.42	25251	1260.67	536.32	0
E08000015	2019	0	2.31	45.37	26869	1279.43	546.34	0
E08000015	2020	2	2.06	81.15	26162	1280.71	651.41	1

E08000015	2021	2	1.95	62.82	25169	1264.98	591.15	0
E08000015	2022	2	2.26	47.53	25856	1272.23	582.58	0
E08000015	2023	2	2.07	45.08	25726	1282.32	624.72	0
E08000016	2016	0	2.28	28.31	22944	733.01	427.84	0
E08000016	2017	0	2.18	32.18	23339	739.46	468.85	0
E08000016	2018	0	2.43	42.05	23957	745.11	357.83	0
E08000016	2019	0	2.58	42.25	24259	750.18	395.31	0
E08000016	2020	0	2.22	78.32	25219	753.84	422.70	1
E08000016	2021	0	2.16	62.93	24545	744.18	432.94	0
E08000016	2022	0	2.68	43.65	23644	748.91	403.86	0
E08000016	2023	0	2.20	43.20	23701	754.96	448.63	0
E08000017	2016	0	2.57	33.08	23160	538.91	493.32	0
E08000017	2017	0	2.63	32.43	22784	543.38	479.91	0
E08000017	2018	0	2.61	42.44	23198	546.20	478.46	0
E08000017	2019	0	2.57	45.94	23659	548.57	484.86	0
E08000017	2020	0	2.15	91.74	26230	550.14	503.59	1
E08000017	2021	0	2.61	76.73	24856	542.98	510.64	0
E08000017	2022	0	2.54	57.23	25133	546.96	507.31	0
E08000017	2023	0	2.69	57.13	24254	552.62	568.52	0
E08000018	2016	0	2.54	32.85	22802	914.13	533.39	0
E08000018	2017	0	2.60	32.60	22646	919.17	487.47	0
E08000018	2018	0	2.42	35.31	22785	923.70	479.53	0
E08000018	2019	0	2.81	44.01	23492	926.28	493.58	0
E08000018	2020	0	2.00	89.12	24004	924.79	522.68	1
E08000018	2021	0	2.47	73.33	22479	928.97	538.75	0
E08000018	2022	0	2.50	54.49	22978	936.25	509.78	0
E08000018	2023	0	2.54	52.30	24005	946.47	545.46	0
E08000019	2016	0	2.43	30.20	23643	1563.95	444.31	0
E08000019	2017	0	2.54	30.11	24443	1570.38	413.61	0
E08000019	2018	0	2.58	27.87	24251	1583.20	442.70	0
E08000019	2019	0	2.75	33.96	24668	1589.58	456.24	0
E08000019	2020	0	2.25	73.38	24874	1601.43	543.86	1
E08000019	2021	2	2.43	69.34	25297	1506.81	596.91	0
E08000019	2022	2	2.71	52.17	24924	1534.81	563.71	0
E08000019	2023	2	2.60	52.30	24446	1558.09	576.00	0
E08000021	2016	0	2.59	33.01	26556	2575.43	501.71	0
E08000021	2017	0	2.61	43.90	25854	2569.91	623.56	0
E08000021	2018	0	2.45	52.47	26745	2607.73	620.31	0
E08000021	2019	0	2.85	54.51	27098	2630.52	679.95	0
E08000021	2020	0	2.27	90.79	27345	2665.30	742.36	1
E08000021	2021	0	2.77	77.10	27217	2591.26	706.10	0
E08000021	2022	0	2.38	55.50	27461	2661.84	664.21	0
E08000021	2023	0	2.36	51.32	25369	2710.25	706.48	0
E08000022	2016	0	2.45	31.56	25144	2396.94	595.22	0
E08000022	2017	0	2.42	32.04	25855	2410.69	608.73	0
E08000022	2018	0	2.57	38.29	25751	2428.52	611.97	0
E08000022	2019	0	2.91	45.54	25610	2451.13	568.05	0
E08000022	2020	0	2.57	79.00	26472	2462.43	875.26	1
E08000022	2021	0	2.55	60.92	25573	2465.79	727.39	0

E08000022	2022	0	2.62	43.47	25327	2482.16	656.32	0
E08000022	2023	0	2.57	42.45	24967	2496.96	675.35	0
E08000023	2016	0	2.44	54.05	24770	2226.52	695.20	0
E08000023	2017	0	2.65	55.55	24213	2228.56	670.48	0
E08000023	2018	0	2.57	63.62	24204	2239.14	655.82	0
E08000023	2019	0	2.67	74.63	22992	2250.88	708.16	0
E08000023	2020	0	2.29	114.42	24825	2253.22	763.90	1
E08000023	2021	0	2.71	93.81	24390	2205.25	771.02	0
E08000023	2022	0	2.78	72.49	22630	2215.58	790.39	0
E08000023	2023	0	2.60	67.02	21267	2225.43	823.25	0
E08000024	2016	0	2.64	38.77	23292	1990.70	629.49	0
E08000024	2017	0	2.69	40.22	23036	1985.60	478.82	0
E08000024	2018	0	2.73	46.67	22399	1986.80	479.93	0
E08000024	2019	0	3.10	61.25	22426	1989.04	508.80	0
E08000024	2020	0	2.50	99.41	23024	1990.04	590.61	1
E08000024	2021	0	1.83	80.41	21359	1964.01	595.57	0
E08000024	2022	0	2.94	58.97	21210	1987.65	584.83	0
E08000024	2023	0	3.00	52.65	21554	2012.98	631.22	0
E08000026	2016	0	2.36	22.97	25106	3577.80	377.22	0
E08000026	2017	0	2.38	22.00	25483	3651.18	368.58	0
E08000026	2018	0	2.58	24.38	26064	3718.46	371.75	0
E08000026	2019	0	2.44	35.75	25732	3766.47	392.07	0
E08000026	2020	0	2.00	78.21	26279	3846.21	373.92	1
E08000026	2021	0	2.00	78.82	26386	3480.57	473.75	0
E08000026	2022	0	2.10	64.52	26613	3577.58	462.34	0
E08000026	2023	0	2.21	66.54	30949	3656.46	513.85	0
E08000027	2016	0	2.53	36.53	25060	3242.54	473.97	0
E08000027	2017	2	2.13	39.65	24669	3260.77	479.15	0
E08000027	2018	2	2.59	48.74	24332	3273.09	476.35	0
E08000027	2019	2	2.66	52.86	26077	3282.99	495.41	0
E08000027	2020	2	1.75	92.04	27700	3290.82	550.59	1
E08000027	2021	1	2.04	74.85	25379	3303.25	545.87	0
E08000027	2022	1	2.35	59.85	24309	3317.04	518.14	0
E08000027	2023	1	2.14	57.86	23194	3334.39	573.35	0
E08000028	2016	0	2.58	44.15	22450	3771.81	416.80	0
E08000028	2017	0	2.65	43.84	22286	3803.93	460.61	0
E08000028	2018	0	2.71	45.64	22124	3826.35	490.24	0
E08000028	2019	0	2.40	58.81	22059	3838.87	527.62	0
E08000028	2020	0	1.73	115.87	22499	3845.79	566.01	1
E08000028	2021	0	2.47	96.45	22555	3994.08	558.34	0
E08000028	2022	0	2.43	78.53	21307	4027.42	590.23	0
E08000028	2023	0	2.00	77.59	21546	4061.72	905.89	0
E08000031	2016	0	2.67	51.62	22891	3695.75	661.75	0
E08000031	2017	0	2.50	52.18	24386	3743.35	668.22	0
E08000031	2018	0	2.57	63.88	22372	3773.34	667.99	0
E08000031	2019	0	2.54	74.11	22994	3792.76	715.49	0
E08000031	2020	0	1.93	128.44	23612	3807.88	771.26	1
E08000031	2021	0	2.33	111.55	24383	3802.54	575.03	0
E08000031	2022	0	2.32	92.15	25706	3858.02	546.49	0

E08000031	2023	0	2.36	91.33	23489	3923.20	598.03	0
E08000032	2016	0	2.25	34.62	24170	1458.11	437.52	0
E08000032	2017	0	2.28	33.81	23285	1459.53	460.32	0
E08000032	2018	0	2.56	41.21	23833	1466.00	466.52	0
E08000032	2019	0	2.67	58.78	24239	1473.11	460.71	0
E08000032	2020	0	1.64	115.76	26506	1479.53	498.61	1
E08000032	2021	0	1.63	102.62	24548	1492.76	510.15	0
E08000032	2022	0	2.12	82.83	24681	1509.32	482.77	0
E08000032	2023	0	2.12	82.07	24673	1528.97	493.87	0
E08000033	2016	2	2.18	25.18	24901	576.35	475.39	0
E08000033	2017	2	2.15	31.04	24085	575.48	487.10	0
E08000033	2018	2	2.18	43.26	24204	577.21	493.09	0
E08000033	2019	2	2.35	45.31	24553	580.98	510.32	0
E08000033	2020	0	2.00	85.22	25411	580.94	555.56	1
E08000033	2021	0	2.08	69.34	24237	568.24	563.75	0
E08000033	2022	0	2.31	50.91	24729	570.56	570.28	0
E08000033	2023	0	2.11	53.00	24087	573.51	617.93	0
E08000034	2016	2	2.30	25.82	23957	1069.75	462.69	0
E08000034	2017	2	2.29	27.78	24273	1069.99	470.22	0
E08000034	2018	2	2.53	39.28	24511	1073.86	479.11	0
E08000034	2019	0	2.52	44.72	24996	1076.45	503.45	0
E08000034	2020	0	2.18	83.90	25106	1080.13	580.70	1
E08000034	2021	2	2.33	69.51	23508	1060.71	602.34	0
E08000034	2022	2	2.43	53.91	24327	1071.58	587.37	0
E08000034	2023	0	2.65	54.04	23376	1081.95	608.83	0
E08000035	2016	0	2.31	27.59	26494	1416.95	497.35	0
E08000035	2017	0	2.55	27.76	26477	1422.58	525.13	0
E08000035	2018	0	2.56	28.63	26010	1430.46	464.07	0
E08000035	2019	0	2.66	39.28	25775	1437.61	485.12	0
E08000035	2020	0	2.29	84.13	27331	1447.85	526.39	1
E08000035	2021	0	2.26	68.94	26992	1466.42	507.33	0
E08000035	2022	0	2.37	52.86	26968	1487.75	478.16	0
E08000035	2023	0	2.58	53.67	25443	1503.38	491.36	0
E08000036	2016	0	2.20	23.28	22783	994.73	459.19	0
E08000036	2017	0	2.31	23.75	23280	1006.41	441.63	0
E08000036	2018	0	2.47	24.69	22821	1018.95	460.13	0
E08000036	2019	0	2.38	35.36	23707	1028.62	476.19	0
E08000036	2020	0	1.69	74.49	24355	1038.31	525.04	1
E08000036	2021	0	2.12	59.25	24616	1044.84	526.33	0
E08000036	2022	0	2.58	43.62	24007	1056.34	535.78	0
E08000036	2023	0	2.35	43.81	22852	1068.45	584.31	0
E08000037	2016	0	2.55	34.27	23973	1399.17	660.44	0
E08000037	2017	0	2.57	36.72	23138	1404.91	651.73	0
E08000037	2018	0	2.72	50.37	23077	1405.53	656.47	0
E08000037	2019	0	2.97	55.16	23929	1402.38	670.64	0
E08000037	2020	0	3.00	93.31	24089	1401.65	725.28	1
E08000037	2021	0	2.23	76.57	24334	1361.51	757.24	0
E08000037	2022	0	2.53	56.07	23759	1373.94	728.41	0
E08000037	2023	0	2.22	54.01	23353	1382.35	745.86	0

E09000002	2016	0	2.43	30.49	25040	5462.00	443.31	0
E09000002	2017	0	2.36	30.18	25046	5574.46	452.30	0
E09000002	2018	0	2.65	37.57	25570	5608.51	479.69	0
E09000002	2019	0	2.57	49.56	24723	5632.53	489.63	0
E09000002	2020	0	3.00	125.95	25480	5666.93	520.26	1
E09000002	2021	0	3.00	106.40	26012	5784.10	505.21	0
E09000002	2022	0	2.50	79.17	24968	5823.95	543.67	0
E09000002	2023	0	3.00	84.80	24094	5883.99	570.29	0
E09000003	2016	1	2.42	17.98	34715	4450.61	437.64	0
E09000003	2017	1	2.52	17.74	35780	4470.44	467.68	0
E09000003	2018	1	2.63	20.34	35728	4520.43	461.85	0
E09000003	2019	1	2.85	29.25	35942	4563.42	439.89	0
E09000003	2020	1	2.67	86.99	36419	4598.62	489.24	1
E09000003	2021	1	2.50	74.32	36765	4479.13	526.54	0
E09000003	2022	1	2.57	55.75	33329	4499.76	525.06	0
E09000003	2023	0	2.64	54.98	33540	4552.52	564.06	0
E09000004	2016	1	2.47	16.25	32516	3807.33	377.31	0
E09000004	2017	1	2.46	17.40	32125	3828.55	413.41	0
E09000004	2018	1	2.50	16.54	32305	3846.19	410.14	0
E09000004	2019	1	2.60	24.74	33817	3862.20	416.06	0
E09000004	2020	1	1.86	70.11	35697	3876.21	465.54	1
E09000004	2021	1	2.50	54.44	33979	3833.32	441.32	0
E09000004	2022	1	2.33	38.35	34125	3852.33	443.51	0
E09000004	2023	1	2.50	39.78	33058	3900.71	468.06	0
E09000005	2016	0	2.76	29.13	29965	7592.73	378.41	0
E09000005	2017	0	2.80	28.15	29372	7612.35	404.44	0
E09000005	2018	0	2.59	27.85	29236	7651.51	408.75	0
E09000005	2019	0	2.63	39.58	31281	7627.82	432.74	0
E09000005	2020	0	2.50	125.59	30717	7580.49	455.51	1
E09000005	2021	0	2.40	102.74	29873	7838.73	449.81	0
E09000005	2022	0	2.42	76.11	32127	7891.78	446.14	0
E09000005	2023	0	2.78	79.98	33042	7968.32	473.60	0
E09000006	2016	1	2.49	13.63	41003	2177.30	371.16	0
E09000006	2017	1	2.50	14.02	41514	2193.97	381.56	0
E09000006	2018	1	2.73	16.04	40538	2205.32	396.41	0
E09000006	2019	1	2.55	23.69	40675	2213.58	424.88	0
E09000006	2020	1	2.43	67.44	43603	2216.39	440.99	1
E09000006	2021	1	2.43	51.74	40051	2196.93	467.02	0
E09000006	2022	1	2.40	36.27	40622	2196.02	461.63	0
E09000006	2023	1	2.70	38.33	39959	2205.85	492.55	0
E09000007	2016	0	2.78	19.09	43762	11298.25	631.57	0
E09000007	2017	0	3.00	18.89	48952	11627.77	581.99	0
E09000007	2018	0	2.25	19.54	56303	12034.62	549.54	0
E09000007	2019	0	2.25	26.01	51671	12392.73	535.96	0
E09000007	2020	0	2.00	67.06	47433	12834.45	545.29	1
E09000007	2021	0	2.33	75.66	40806	9660.41	715.89	0
E09000007	2022	0	2.50	53.34	41295	9975.76	709.42	0
E09000007	2023	0	2.00	51.73	42258	10143.13	723.99	0
E09000008	2016	0	2.83	26.30	30936	4419.98	485.70	0

E09000008	2017	0	2.69	45.44	31777	4449.27	513.15	0
E09000008	2018	0	2.57	49.02	32766	4455.15	617.34	0
E09000008	2019	0	2.71	53.09	32202	4470.92	626.29	0
E09000008	2020	0	1.89	112.37	32675	4492.64	612.62	1
E09000008	2021	0	2.53	92.17	33095	4515.11	542.93	0
E09000008	2022	0	2.66	68.97	31612	4539.01	507.46	0
E09000008	2023	2	2.62	70.83	32527	4599.96	519.90	0
E09000009	2016	0	2.53	27.01	32940	6178.77	473.88	0
E09000009	2017	0	2.54	27.89	32389	6170.49	520.78	0
E09000009	2018	0	2.21	33.34	36190	6156.92	478.04	0
E09000009	2019	0	2.13	42.95	34820	6153.75	464.17	0
E09000009	2020	0	2.29	117.51	39780	6127.63	429.89	1
E09000009	2021	0	2.24	91.02	37261	6591.89	396.61	0
E09000009	2022	0	2.38	66.92	34422	6663.66	420.20	0
E09000009	2023	0	2.50	68.76	35504	6757.76	445.48	0
E09000010	2016	0	2.59	27.32	30035	4031.56	474.05	0
E09000010	2017	0	2.65	26.98	30042	4047.50	484.68	0
E09000010	2018	0	2.65	33.46	29770	4061.66	486.67	0
E09000010	2019	0	2.53	42.39	30072	4060.74	490.94	0
E09000010	2020	0	2.20	110.82	29287	4058.71	544.01	1
E09000010	2021	0	2.63	99.50	29105	4010.22	550.75	0
E09000010	2022	0	2.64	75.49	28402	3983.28	560.89	0
E09000010	2023	0	2.54	76.26	31191	3984.79	589.96	0
E09000011	2016	0	2.54	26.05	38041	5546.30	602.69	0
E09000011	2017	0	2.75	26.96	36435	5607.42	556.72	0
E09000011	2018	0	2.53	26.91	39306	5673.57	525.27	0
E09000011	2019	0	2.47	37.30	38536	5708.38	536.12	0
E09000011	2020	0	2.00	98.78	39585	5729.47	546.08	1
E09000011	2021	0	2.50	82.30	35738	5733.83	554.05	0
E09000011	2022	0	2.55	60.30	34085	5786.73	525.81	0
E09000011	2023	0	2.25	62.00	35607	5830.15	553.94	0
E09000014	2016	0	2.25	29.43	31102	9407.64	516.60	0
E09000014	2017	0	2.54	30.37	33778	9163.47	480.40	0
E09000014	2018	0	2.70	31.06	33208	9143.20	528.60	0
E09000014	2019	0	2.63	43.08	33468	9076.41	540.22	0
E09000014	2020	0	2.00	132.44	32557	8996.67	584.29	1
E09000014	2021	0	2.25	114.80	31117	8921.45	612.89	0
E09000014	2022	0	2.33	85.51	34357	8865.78	594.69	0
E09000014	2023	0	3.00	85.50	32950	8879.73	559.55	0
E09000015	2016	0	2.64	14.77	32604	4929.36	408.17	0
E09000015	2017	0	2.71	14.74	33428	4931.90	431.81	0
E09000015	2018	0	2.58	16.53	33663	4957.05	446.23	0
E09000015	2019	0	2.75	24.87	33896	4977.08	463.63	0
E09000015	2020	0	2.60	84.64	37453	5000.33	492.55	1
E09000015	2021	0	2.67	65.97	33640	5171.72	514.66	0
E09000015	2022	0	2.86	45.02	31035	5190.56	489.39	0
E09000015	2023	1	3.07	46.77	30631	5220.49	490.49	0
E09000016	2016	2	2.63	19.85	34274	2208.54	412.17	0
E09000016	2017	2	2.69	20.07	31284	2236.98	395.98	0

E09000016	2018	2	2.79	22.39	30549	2252.46	386.36	0
E09000016	2019	2	2.57	30.47	32972	2267.68	395.69	0
E09000016	2020	2	2.33	79.96	33347	2277.28	472.95	1
E09000016	2021	2	2.44	62.34	32341	2289.26	487.48	0
E09000016	2022	2	2.50	45.69	29990	2312.43	434.34	0
E09000016	2023	2	2.69	49.67	31530	2342.92	482.08	0
E09000018	2016	0	2.68	19.85	33749	4791.67	291.87	0
E09000018	2017	0	2.36	35.47	33636	4755.64	354.58	0
E09000018	2018	0	2.38	39.77	30848	4785.37	324.33	0
E09000018	2019	0	2.56	43.97	32174	4798.46	377.56	0
E09000018	2020	0	2.38	111.07	33380	4802.40	386.82	1
E09000018	2021	0	2.40	87.02	30573	5088.20	382.57	0
E09000018	2022	0	2.67	60.40	31548	5146.18	426.11	0
E09000018	2023	0	3.00	61.50	29390	5225.50	450.51	0
E09000022	2016	0	2.65	28.85	32939	12033.69	493.11	0
E09000022	2017	0	2.79	30.69	37810	11891.96	458.09	0
E09000022	2018	0	2.65	37.22	39709	11960.55	521.89	0
E09000022	2019	0	2.50	43.73	37093	11964.84	466.52	0
E09000022	2020	0	2.43	106.86	40693	11815.03	497.35	1
E09000022	2021	0	2.70	89.52	39068	11656.61	499.19	0
E09000022	2022	0	2.75	64.74	40120	11627.41	529.59	0
E09000022	2023	0	3.00	66.30	45187	11590.82	575.22	0
E09000023	2016	0	2.73	29.94	32924	8547.35	429.28	0
E09000023	2017	0	2.79	31.35	32395	8531.49	439.81	0
E09000023	2018	0	2.62	34.58	33376	8594.61	462.81	0
E09000023	2019	0	2.78	45.62	33093	8659.90	479.36	0
E09000023	2020	0	2.50	112.61	36970	8645.56	581.62	1
E09000023	2021	0	2.50	96.59	32819	8489.84	435.08	0
E09000023	2022	0	2.50	72.20	31879	8466.80	545.50	0
E09000023	2023	0	3.00	75.42	32041	8458.63	544.59	0
E09000024	2016	0	2.47	18.82	49547	5449.31	463.29	0
E09000024	2017	0	2.78	21.84	40446	5476.50	429.82	0
E09000024	2018	0	2.83	26.99	42637	5480.06	405.18	0
E09000024	2019	0	2.76	31.88	40793	5489.69	403.00	0
E09000024	2020	0	2.75	87.18	44075	5487.26	411.18	1
E09000024	2021	0	2.77	64.67	38092	5723.04	410.86	0
E09000024	2022	0	2.65	48.15	36889	5717.54	411.50	0
E09000024	2023	0	2.86	51.61	38848	5720.25	455.45	0
E09000025	2016	0	2.27	23.53	26011	8838.65	406.09	0
E09000025	2017	0	2.73	23.74	25696	9020.56	431.94	0
E09000025	2018	0	2.45	26.28	26537	9124.48	419.04	0
E09000025	2019	0	2.38	37.21	27904	9153.75	418.29	0
E09000025	2020	0	2.00	121.88	28854	9209.55	500.85	1
E09000025	2021	0	2.00	113.42	31291	9089.27	482.58	0
E09000025	2022	0	2.45	77.49	27617	9256.96	446.82	0
E09000025	2023	0	3.00	81.44	29374	9398.43	483.59	0
E09000026	2016	0	2.56	17.72	34955	5301.87	408.45	0
E09000026	2017	0	2.80	16.29	34094	5346.80	398.51	0
E09000026	2018	0	2.83	18.97	32991	5383.52	393.28	0

E09000026	2019	0	2.63	29.67	34219	5407.69	391.99	0
E09000026	2020	0	2.50	97.63	34166	5416.18	450.50	1
E09000026	2021	0	2.50	82.71	33069	5490.21	432.56	0
E09000026	2022	0	2.80	58.29	30505	5519.37	405.90	0
E09000026	2023	0	2.63	63.05	31914	5553.54	425.55	0
E09000028	2016	0	2.71	27.51	35366	10470.98	517.04	0
E09000028	2017	0	2.89	39.13	36460	10504.71	442.54	0
E09000028	2018	0	2.50	43.81	35631	10605.80	424.90	0
E09000028	2019	0	2.77	45.62	39554	10658.42	444.47	0
E09000028	2020	0	2.00	103.56	40792	10694.65	485.52	1
E09000028	2021	0	2.20	87.93	38197	10238.72	500.37	0
E09000028	2022	0	2.11	61.94	39147	10410.26	419.45	0
E09000028	2023	0	3.00	61.14	39863	10544.33	477.52	0
E09000031	2016	0	2.59	27.72	29143	7107.90	410.53	0
E09000031	2017	0	2.84	27.21	30435	7099.19	404.05	0
E09000031	2018	0	2.86	29.74	30346	7129.98	419.25	0
E09000031	2019	0	2.58	39.60	32311	7137.27	430.87	0
E09000031	2020	0	2.00	119.84	34736	7135.75	476.33	1
E09000031	2021	0	2.67	96.71	33007	7164.35	469.02	0
E09000031	2022	0	2.89	68.89	32733	7119.98	476.48	0
E09000031	2023	0	2.92	76.94	32048	7111.01	497.84	0
E09000032	2016	1	2.30	16.85	52281	8974.86	444.77	0
E09000032	2017	1	2.63	17.17	51775	9178.18	446.28	0
E09000032	2018	1	2.81	19.14	54344	9269.52	427.13	0
E09000032	2019	1	2.44	24.44	51506	9360.46	408.90	0
E09000032	2020	1	3.00	70.18	52658	9363.21	419.63	1
E09000032	2021	1	3.00	55.59	53004	9324.36	418.02	0
E09000032	2022	1	2.89	40.19	54322	9351.36	413.96	0
E09000032	2023	0	3.00	41.18	48188	9412.08	440.68	0
E09000033	2016	1	2.50	17.39	54650	11239.83	587.24	0
E09000033	2017	1	2.56	17.98	75662	11111.91	558.17	0
E09000033	2018	1	2.33	19.35	73117	11589.81	583.29	0
E09000033	2019	1	2.40	24.94	63946	11861.84	428.85	0
E09000033	2020	1	2.40	63.96	56459	12244.79	428.12	1
E09000033	2021	1	2.00	71.03	53338	9306.15	549.47	0
E09000033	2022	1	2.25	50.18	48416	9526.37	510.55	0
E09000033	2023	0	2.50	48.32	49571	9597.52	537.61	0
E10000003	2016	2	2.69	8.30	32450	213.47	420.39	0
E10000003	2017	2	2.84	8.88	31042	212.26	421.00	0
E10000003	2018	1	2.73	10.86	31710	213.32	438.80	0
E10000003	2019	1	2.79	17.61	31242	213.99	455.24	0
E10000003	2020	1	2.33	49.40	31869	215.19	540.09	1
E10000003	2021	2	2.82	35.90	30899	222.79	504.24	0
E10000003	2022	2	2.44	27.22	29855	225.98	496.82	0
E10000003	2023	2	2.25	26.68	29937	229.31	489.08	0
E10000007	2016	0	2.60	15.63	26360	308.05	486.18	0
E10000007	2017	0	2.59	16.98	25828	310.48	475.62	0
E10000007	2018	1	2.64	21.72	25826	312.12	495.65	0
E10000007	2019	1	2.59	27.22	25687	314.69	508.81	0

E10000007	2020	1	1.97	57.92	27264	316.45	579.03	1
E10000007	2021	1	2.20	43.36	26242	312.40	622.31	0
E10000007	2022	1	2.42	32.62	24832	315.07	637.71	0
E10000007	2023	1	2.59	34.10	25204	318.97	582.32	0
E10000008	2016	1	2.76	11.29	23547	117.51	456.06	0
E10000008	2017	1	2.64	11.48	23079	118.62	460.49	0
E10000008	2018	1	2.70	13.08	22814	119.84	479.04	0
E10000008	2019	1	2.79	19.60	23338	120.96	499.42	0
E10000008	2020	1	2.07	56.01	23090	122.22	603.09	1
E10000008	2021	1	2.33	38.33	23225	122.78	577.22	0
E10000008	2022	1	2.53	24.90	22926	124.50	575.65	0
E10000008	2023	1	2.70	24.47	22857	125.66	603.57	0
E10000011	2016	2	2.55	16.86	26104	317.53	561.90	0
E10000011	2017	2	2.51	21.48	25672	320.12	563.95	0
E10000011	2018	1	2.60	28.68	25314	321.47	565.82	0
E10000011	2019	1	2.70	33.37	26442	323.25	559.23	0
E10000011	2020	1	2.11	77.09	27410	324.19	662.62	1
E10000011	2021	1	2.48	61.66	26446	317.28	659.22	0
E10000011	2022	1	2.57	45.56	26129	319.72	634.88	0
E10000011	2023	1	2.66	42.47	24848	322.90	686.88	0
E10000012	2016	1	2.59	16.40	31967	393.83	500.93	0
E10000012	2017	1	2.50	16.54	31505	397.31	491.39	0
E10000012	2018	1	2.62	21.01	31230	399.90	488.18	0
E10000012	2019	1	2.64	27.38	31035	402.84	485.57	0
E10000012	2020	1	2.15	70.08	30915	405.17	544.31	1
E10000012	2021	1	2.34	51.41	30528	407.50	538.96	0
E10000012	2022	1	2.32	34.96	30754	411.17	511.10	0
E10000012	2023	1	2.36	35.37	31117	417.29	522.95	0
E10000013	2016	2	2.49	13.54	27384	230.40	427.07	0
E10000013	2017	2	2.74	12.86	26891	232.25	415.58	0
E10000013	2018	1	2.82	19.14	27790	234.26	422.60	0
E10000013	2019	1	2.85	23.98	27985	235.56	445.33	0
E10000013	2020	1	2.50	57.57	27498	236.90	489.79	1
E10000013	2021	1	2.58	40.95	27004	239.12	474.00	0
E10000013	2022	1	2.49	27.78	28565	241.45	472.95	0
E10000013	2023	1	2.45	29.81	26670	244.71	512.51	0
E10000014	2016	1	2.65	9.29	30947	363.95	409.18	0
E10000014	2017	1	2.73	10.54	31879	366.71	397.90	0
E10000014	2018	1	2.70	12.40	31528	368.21	414.09	0
E10000014	2019	1	2.77	17.75	31081	369.93	425.50	0
E10000014	2020	1	2.36	52.00	30700	371.72	479.22	1
E10000014	2021	1	2.49	38.01	30141	376.27	480.86	0
E10000014	2022	1	2.56	27.90	29568	379.41	467.37	0
E10000014	2023	1	2.33	27.65	29251	383.66	489.43	0
E10000015	2016	1	2.57	13.24	34972	716.17	442.95	0
E10000015	2017	1	2.65	14.10	35283	718.74	436.58	0
E10000015	2018	1	2.65	17.23	36593	720.83	445.31	0
E10000015	2019	1	2.63	22.28	34849	723.96	487.53	0
E10000015	2020	1	2.00	61.84	36834	727.71	572.94	1

E10000015	2021	1	2.34	46.56	32760	730.72	551.93	0
E10000015	2022	1	2.23	31.57	34417	733.73	491.16	0
E10000015	2023	1	2.47	32.57	34509	743.08	511.70	0
E10000016	2016	1	2.55	19.79	29019	423.65	403.76	0
E10000016	2017	1	2.43	21.52	29163	427.15	403.57	0
E10000016	2018	1	2.50	25.79	29064	430.99	409.96	0
E10000016	2019	1	2.64	33.03	30779	434.83	431.99	0
E10000016	2020	1	2.08	74.62	31096	436.89	493.49	1
E10000016	2021	1	2.31	57.44	31141	433.99	507.58	0
E10000016	2022	1	2.32	40.97	28668	438.49	487.02	0
E10000016	2023	1	2.34	41.31	28178	444.74	517.99	0
E10000017	2016	2	2.49	23.29	25805	388.84	523.45	0
E10000017	2017	2	2.55	26.41	25076	389.83	530.43	0
E10000017	2018	1	2.59	30.12	25545	392.49	536.25	0
E10000017	2019	1	2.63	36.33	25953	395.67	538.75	0
E10000017	2020	1	2.05	73.39	25403	398.03	597.70	1
E10000017	2021	1	2.41	57.37	24676	401.12	594.74	0
E10000017	2022	1	2.41	43.49	24542	407.50	583.22	0
E10000017	2023	1	2.54	43.79	25362	413.80	561.96	0
E10000018	2016	1	2.63	9.92	26708	327.75	401.91	0
E10000018	2017	1	2.66	11.31	26269	331.23	395.73	0
E10000018	2018	1	2.55	15.17	26013	335.10	385.99	0
E10000018	2019	1	2.65	19.43	26664	338.88	399.23	0
E10000018	2020	1	1.80	49.96	27925	342.21	446.42	1
E10000018	2021	1	2.33	36.77	25808	341.96	457.68	0
E10000018	2022	1	2.37	26.69	26441	347.05	440.73	0
E10000018	2023	1	2.36	26.63	26043	352.36	453.95	0
E10000019	2016	2	2.51	19.65	23889	121.82	409.52	0
E10000019	2017	2	2.61	20.73	23942	123.09	439.79	0
E10000019	2018	1	2.65	25.68	23877	123.85	450.78	0
E10000019	2019	1	2.70	33.39	24198	124.74	462.52	0
E10000019	2020	1	2.11	68.37	24524	125.58	452.75	1
E10000019	2021	1	2.26	52.87	24230	126.09	464.67	0
E10000019	2022	1	2.15	39.37	24107	127.17	465.58	0
E10000019	2023	1	2.40	37.92	24274	128.47	467.35	0
E10000020	2016	2	2.56	16.19	23566	162.09	539.05	0
E10000020	2017	2	2.31	18.64	22547	163.10	551.75	0
E10000020	2018	1	2.47	20.48	23594	164.06	562.16	0
E10000020	2019	1	2.43	28.13	23587	164.83	579.45	0
E10000020	2020	1	1.82	63.61	24256	165.97	641.09	1
E10000020	2021	1	2.09	47.95	24598	166.75	612.66	0
E10000020	2022	1	2.22	34.46	24014	168.05	617.18	0
E10000020	2023	1	2.11	34.47	23706	169.46	652.58	0
E10000024	2016	0	2.54	18.77	27047	388.48	493.93	0
E10000024	2017	0	2.47	19.86	26290	391.90	507.62	0
E10000024	2018	2	2.48	21.23	26800	394.43	512.15	0
E10000024	2019	2	2.57	29.30	27151	396.87	506.76	0
E10000024	2020	2	2.03	61.94	26937	399.34	564.34	1
E10000024	2021	1	2.27	48.40	25594	395.93	563.60	0

E10000024	2022	1	2.50	36.97	24734	400.30	537.52	0
E10000024	2023	1	2.48	37.47	24250	405.74	574.65	0
E10000025	2016	2	2.73	7.80	32971	262.16	435.64	0
E10000025	2017	2	2.63	8.78	31019	261.88	413.15	0
E10000025	2018	2	2.81	14.21	31606	263.83	445.51	0
E10000025	2019	2	2.85	17.52	31610	265.42	457.22	0
E10000025	2020	2	2.45	47.85	31979	267.42	521.56	1
E10000025	2021	2	2.71	34.79	30903	278.80	492.18	0
E10000025	2022	2	2.74	25.23	30292	283.55	510.60	0
E10000025	2023	2	2.31	25.26	29250	288.53	448.62	0
E10000028	2016	1	2.36	11.82	26297	330.54	475.73	0
E10000028	2017	1	2.27	13.43	26076	331.96	432.69	0
E10000028	2018	1	2.43	16.97	26396	333.63	443.30	0
E10000028	2019	1	2.58	24.62	26259	335.29	420.59	0
E10000028	2020	1	2.07	59.10	26990	336.66	473.73	1
E10000028	2021	1	2.48	45.63	25711	334.64	509.37	0
E10000028	2022	1	2.49	34.22	25710	337.93	467.06	0
E10000028	2023	1	2.18	35.24	25970	342.27	513.87	0
E10000029	2016	1	2.37	16.17	26881	193.40	463.81	0
E10000029	2017	1	2.69	18.65	26964	196.44	485.92	0
E10000029	2018	1	2.81	23.21	26785	196.84	488.50	0
E10000029	2019	1	2.73	28.70	27055	197.63	497.86	0
E10000029	2020	1	2.07	63.17	27088	197.60	614.57	1
E10000029	2021	1	2.13	48.40	26017	198.15	569.30	0
E10000029	2022	1	2.57	34.73	25147	199.56	546.84	0
E10000029	2023	1	2.55	34.51	25505	201.95	578.70	0
E10000030	2016	1	2.64	7.61	36845	704.49	502.50	0
E10000030	2017	1	2.56	8.58	38077	709.74	496.17	0
E10000030	2018	1	2.61	8.98	38418	712.50	502.10	0
E10000030	2019	1	2.68	13.35	40177	716.28	506.21	0
E10000030	2020	1	2.17	49.02	39658	718.45	570.97	1
E10000030	2021	1	2.41	35.87	37900	721.89	559.31	0
E10000030	2022	1	2.60	25.61	38452	729.26	549.57	0
E10000030	2023	1	2.45	26.03	38075	739.52	532.43	0
E10000031	2016	2	2.61	12.90	30258	281.54	391.41	0
E10000031	2017	2	2.69	16.29	29897	285.49	398.56	0
E10000031	2018	1	2.51	21.12	30007	288.75	411.19	0
E10000031	2019	1	2.59	25.65	31090	292.25	417.63	0
E10000031	2020	1	2.25	59.12	29792	295.21	460.22	1
E10000031	2021	1	2.33	44.26	29720	302.98	441.03	0
E10000031	2022	1	2.47	32.78	28619	307.82	462.48	0
E10000031	2023	1	2.69	31.34	28326	313.71	507.96	0
E10000032	2016	1	2.68	12.10	28596	416.99	458.41	0
E10000032	2017	1	2.57	12.72	28313	421.23	469.25	0
E10000032	2018	1	2.43	15.31	29299	424.45	491.71	0
E10000032	2019	1	2.48	22.38	28937	426.76	508.64	0
E10000032	2020	1	1.86	65.28	29344	428.57	539.02	1
E10000032	2021	1	2.37	48.09	27548	437.17	548.29	0
E10000032	2022	1	2.53	33.29	27616	441.19	511.78	0

E10000032	2023	1	2.58	32.98	27360	446.47	533.29	0
E10000034	2016	1	2.87	15.33	26109	334.99	412.94	0
E10000034	2017	1	2.49	16.27	26119	338.04	429.02	0
E10000034	2018	1	2.63	20.20	25799	340.16	429.44	0
E10000034	2019	1	2.59	27.35	26378	342.30	438.32	0
E10000034	2020	1	2.31	65.50	25964	343.62	486.65	1
E10000034	2021	1	2.47	49.32	26580	347.57	491.68	0
E10000034	2022	1	2.43	38.15	26184	350.21	483.72	0
E10000034	2023	1	2.38	36.69	26020	353.71	504.58	0

Table C2. Code

In [ ]:

```
# Part 1 — Preprocess + Rule-R classification + Variable-wise Linear vs
Quadratic
# Scope: SEED=42; 2016-2023; t=0..7 centered; for identifying growth
function shape
# Output root: F:/文档/UB/MGRCM0042/Data/Analysis/Result/

# %% 0) Imports
import os, warnings, gc
from typing import List, Dict, Tuple
import numpy as np
import pandas as pd
import arviz as az
import xarray as xr

import jax
import jax.numpy as jnp
from jax import random

import numpyro
import numpyro.distributions as dist
from numpyro.infer import MCMC, NUTS

try:
    from numpyro.infer import log_likelihood
except ImportError:
    from numpyro.infer.util import log_likelihood

warnings.filterwarnings("ignore")
numpyro.enable_x64()

import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (9,5)
plt.rcParams["figure.dpi"] = 120
plt.rcParams["font.family"] = "Times New Roman"
```

```
# %% 1) Config
DATA_PATH = r"F:/文档/UB/MGRCM0042/Data/Analysis/Full_Data_Final.xlsx"
OUT_ROOT  = r"F:/文档/UB/MGRCM0042/Data/Analysis/Result"
DIR_PREP  = os.path.join(OUT_ROOT, "prep")
DIR_CMP   = os.path.join(OUT_ROOT, "comparisons")
os.makedirs(DIR_PREP, exist_ok=True)
os.makedirs(DIR_CMP, exist_ok=True)

YEAR_START, YEAR_END = 2016, 2023
SEED = 42
CHAINS = 5
WARMUP = 3000
SAMPLES = 2000
TARGET_ACCEPT = 0.99

# Required columns
REQUIRED = ["ONSCode","Year","Plan_P",
          "Overall","LATE_P18","Mean","Unemp_Rate","P_Density"]

# Party encoding
PARTY_MAP = {0:"LAB", 1:"CON", 2:"NOC"}
PARTY_LIST = ["LAB","CON","NOC"]

# Variables to compare (original units)
VAR_LIST = [
    ("Overall",   "Overall"),
    ("LATE_P18",  "LATE_P18"),
    ("Mean",      "Mean"),
    ("Unemp_Rate","Unemp_Rate"),
    ("P_Density", "P_Density"),
]

# %% 2) Load & sanity checks
df_raw = pd.read_excel(DATA_PATH)
df_raw.columns = [c.strip() for c in df_raw.columns]

missing = [c for c in REQUIRED if c not in df_raw.columns]
if missing:
    raise ValueError(f"Missing required columns: {missing}")

for c in REQUIRED:
    if c != "ONSCode":
        df_raw[c] = pd.to_numeric(df_raw[c], errors="coerce")

# Filter years, remove duplicates, sort
df = (df_raw
      .loc[(df_raw["Year"]>=YEAR_START) & (df_raw["Year"]<=YEAR_END)]
      .dropna(subset=["ONSCode","Year"])
```

```
      .copy())
if df.duplicated(subset=["ONSCode","Year"]).any():
    df = df[~df.duplicated(subset=["ONSCode","Year"],
keep="first")].copy()
df = df.sort_values(["ONSCode","Year"]).reset_index(drop=True)

# Check continuous years and balanced panel
years = sorted(df["Year"].unique().tolist())
expected = list(range(YEAR_START, YEAR_END+1))
if years != expected:
    raise AssertionError(f"Years must be continuous {expected}, got
{years}")
counts = df.groupby("ONSCode")["Year"].nunique()
if not (counts==len(expected)).all():
    bad = counts[counts!=len(expected)]
    raise AssertionError(f"Unbalanced panel: LAs with insufficient
years: {bad.to_dict()}")

# ===== 2.5) Exclude "reverter" LAs (changed during period but
first==last) =====
first_plan = df.groupby("ONSCode")["Plan_P"].first()
last_plan  = df.groupby("ONSCode")["Plan_P"].last()
nuniq_plan = df.groupby("ONSCode")["Plan_P"].nunique()

reverters_idx = nuniq_plan[(nuniq_plan>1) &
(first_plan==last_plan)].index
N_before = df["ONSCode"].nunique()
df = df[~df["ONSCode"].isin(reverters_idx)].copy()
N_after = df["ONSCode"].nunique()
print(f"[Part-1] Excluded reverters: n={len(reverters_idx)} | Remaining
LAs: N={N_after} (from {N_before})")

# Check balance after exclusion
counts2 = df.groupby("ONSCode")["Year"].nunique()
assert (counts2==len(expected)).all(), "Unbalanced after exclusion,
check data."

# Time variables: t=0..7 and centered (set after exclusion)
df["t"] = df["Year"] - YEAR_START            # 0..7
t_vals = np.array(sorted(df["t"].unique()), dtype=float)
t_centered = t_vals - t_vals.mean()          # -3.5..+3.5
df["t_centered"] = df["t"].map(dict(zip(t_vals, t_centered)))

# Export basic cleaned results (sample after reverter exclusion)
df.to_csv(os.path.join(DIR_PREP, "clean_panel_basic.csv"), index=False,
encoding="utf-8-sig")
```

```
# %% 3) Rule R: denoising + endpoint determination + party switch +
(non-switchers) long-term party≥5/8 or NMOT
def seq_to_parties(arr: np.ndarray) -> List[str]:
    return [PARTY_MAP.get(int(x), "UNK") for x in arr]

def denoise_single_islands(seq: List[str]) -> List[str]:
    """Fill single-year 'islands': ... A, B, A ... -> ... A, A, A ...;
iterative until convergence."""
    changed = True
    seq = seq.copy()
    while changed:
        changed = False
        for i in range(1, len(seq)-1):
            if seq[i-1] == seq[i+1] and seq[i] != seq[i-1]:
                seq[i] = seq[i-1]
                changed = True
    return seq

def run_lengths(seq: List[str]) -> List[Tuple[str,int,int,int]]:
    """Return each run as (party, start_index, end_index, length)."""
    runs = []
    if not seq: return runs
    cur = seq[0]; start = 0
    for i in range(1, len(seq)):
        if seq[i] != cur:
            runs.append((cur, start, i-1, i-1-start+1))
            cur, start = seq[i], i
    runs.append((cur, start, len(seq)-1, len(seq)-start))
    return runs

def classify_by_rule_R(plan_seq: List[str], min_end_run=2) -> Dict[str,
str]:
    """Denoise → endpoint parties (first/last segments ≥2 years for
clarity) → party switch (X!=Y)."""
    denoised = denoise_single_islands(plan_seq)
    runs = run_lengths(denoised)
    if not runs:
        return
{"X":"UNCLEAR","Y":"UNCLEAR","Switch":"NO","Direction":"","DenoisedSeq"
:""}
    first_party, _, _, L1 = runs[0]
    last_party,  _, _, Lk = runs[-1]
    X = first_party if L1 >= min_end_run else "UNCLEAR"
    Y = last_party  if Lk >= min_end_run else "UNCLEAR"
    if (X!="UNCLEAR") and (Y!="UNCLEAR") and (X!=Y):
        return {"X":X, "Y":Y, "Switch":"YES", "Direction":f"{X}->{Y}",
"DenoisedSeq":"-".join(denoised)}
    else:
```

```
        return {"X":X, "Y":Y, "Switch":"NO", "Direction":"",
"DenoisedSeq":"-".join(denoised)}

def longrun_group_majority_5of8(denoised_seq: List[str]) -> str:
    """Party with ≥5 years out of 8 → that party; otherwise NMOT."""
    counts = {p: denoised_seq.count(p) for p in PARTY_LIST}
    winner = max(counts, key=counts.get)
    return winner if counts[winner] >= 5 else "NMOT"

# Audit table (based on sample after reverter exclusion)
rows = []
for la, g in df.groupby("ONSCode", sort=False):
    seq_raw = seq_to_parties(g["Plan_P"].values.astype(int))
    R = classify_by_rule_R(seq_raw, min_end_run=2)
    den_seq = R["DenoisedSeq"].split("-") if R["DenoisedSeq"] else []
    longrun = ""
    if R["Switch"]=="NO" and (R["X"] == R["Y"]) and
(R["X"]!="UNCLEAR"):
        longrun = longrun_group_majority_5of8(den_seq)
    rows.append({
        "ONSCode": la,
        "PlanSeq_raw": "-".join(seq_raw),
        "PlanSeq_denoised": R["DenoisedSeq"],
        "X_party": R["X"],
        "Y_party": R["Y"],
        "Switch": R["Switch"],
        "Direction": R["Direction"],
        "Longrun_Group_for_A": longrun,
        "Assigned_Model": "B" if R["Switch"]=="YES" else "A"
    })

audit = pd.DataFrame(rows)
audit.to_csv(os.path.join(DIR_PREP, "audit_classification.csv"),
index=False, encoding="utf-8-sig")

# Count groups
A_mask = (audit["Assigned_Model"]=="A")
B_mask = (audit["Assigned_Model"]=="B")

A_counts = (
    audit.loc[A_mask, "Longrun_Group_for_A"]
         .replace({"": "NMOT"})
         .value_counts()
         .reindex(["LAB","CON","NOC","NMOT"], fill_value=0)
)
B_dirs =
["LAB->CON","LAB->NOC","CON->LAB","CON->NOC","NOC->LAB","NOC->CON"]
```

```
B_counts = audit.loc[B_mask,
"Direction"].value_counts().reindex(B_dirs, fill_value=0)

group_counts = pd.concat([

A_counts.rename_axis("Group").reset_index(name="Count").assign(Set="A_l
ongrun"),

B_counts.rename_axis("Group").reset_index(name="Count").assign(Set="B_t
ransition"),
], ignore_index=True)
group_counts.to_csv(os.path.join(DIR_PREP, "group_counts.csv"),
index=False, encoding="utf-8-sig")

print("Saved:", os.path.join(DIR_PREP, "clean_panel_basic.csv"))
print("Saved:", os.path.join(DIR_PREP, "audit_classification.csv"))
print("Saved:", os.path.join(DIR_PREP, "group_counts.csv"))

# %% 4) Panel expansion (by variable; original units; no TVC)
def build_panel_for(col: str):
    yrs = sorted(df["Year"].unique())
    las = df["ONSCode"].astype("category").cat.categories.tolist()
    wide = (df.pivot_table(index="ONSCode", columns="Year", values=col,
aggfunc="mean")
              .reindex(index=las, columns=yrs))
    if wide.isna().any().any():
        raise ValueError(f"{col} expansion has missing values.")
    Y = jnp.asarray(wide.values.astype(float))
    # Centered time vector (based on post-exclusion sample)
    t_vals = np.array(sorted(df["t_centered"].unique()), dtype=float)
    t = jnp.asarray(t_vals); t2 = t**2
    return Y, t, t2, yrs, las

# %% 5) Variable-wise: linear/quadratic growth models (hierarchical
intercepts/slopes only)
def model_var_linear(Y, t):
    N, T = Y.shape
    mu_a = numpyro.sample("mu_alpha", dist.Normal(0.,1.))
    sd_a = numpyro.sample("sd_alpha", dist.HalfNormal(1.))
    mu_b = numpyro.sample("mu_beta",  dist.Normal(0.,1.))
    sd_b = numpyro.sample("sd_beta",  dist.HalfNormal(1.))
    with numpyro.plate("la", N):
        z_a = numpyro.sample("z_alpha", dist.Normal(0.,1.))
        z_b = numpyro.sample("z_beta",  dist.Normal(0.,1.))
        alpha = numpyro.deterministic("alpha", mu_a + z_a*sd_a)
        beta  = numpyro.deterministic("beta",  mu_b + z_b*sd_b)
    mu = alpha[:,None] + beta[:,None]*t[None,:]
    sigma = numpyro.sample("sigma_y", dist.HalfNormal(1.))
```

```
    numpyro.sample("Y_obs", dist.Normal(mu, sigma), obs=Y)

def model_var_quadratic(Y, t, t2):
    N, T = Y.shape
    mu_a  = numpyro.sample("mu_alpha",  dist.Normal(0.,1.))
    sd_a  = numpyro.sample("sd_alpha",  dist.HalfNormal(1.))
    mu_b  = numpyro.sample("mu_beta",   dist.Normal(0.,1.))
    sd_b  = numpyro.sample("sd_beta",   dist.HalfNormal(1.))
    mu_b2 = numpyro.sample("mu_beta2",  dist.Normal(0.,1.))
    sd_b2 = numpyro.sample("sd_beta2",  dist.HalfNormal(1.))
    with numpyro.plate("la", N):
        z_a  = numpyro.sample("z_alpha",  dist.Normal(0.,1.))
        z_b  = numpyro.sample("z_beta",   dist.Normal(0.,1.))
        z_b2 = numpyro.sample("z_beta2",  dist.Normal(0.,1.))
        alpha = numpyro.deterministic("alpha",  mu_a  + z_a*sd_a)
        beta  = numpyro.deterministic("beta",   mu_b  + z_b*sd_b)
        beta2 = numpyro.deterministic("beta2",  mu_b2 + z_b2*sd_b2)
    mu = alpha[:,None] + beta[:,None]*t[None,:] +
beta2[:,None]*t2[None,:]
    sigma = numpyro.sample("sigma_y", dist.HalfNormal(1.))
    numpyro.sample("Y_obs", dist.Normal(mu, sigma), obs=Y)

# %% 6) Run one variable's model set + compute LOO / LOOIC / BIC-like
def run_one(var_label: str, Y, t, t2, quadratic: bool):
    rng_key = random.PRNGKey(SEED)
    model_fn = model_var_quadratic if quadratic else model_var_linear
    kernel = NUTS(model_fn, target_accept_prob=TARGET_ACCEPT)
    mcmc = MCMC(kernel, num_warmup=WARMUP, num_samples=SAMPLES,
num_chains=CHAINS, progress_bar=True)
    if quadratic:
        mcmc.run(rng_key, Y=Y, t=t, t2=t2)
        model_kwargs = dict(Y=Y, t=t, t2=t2)
    else:
        mcmc.run(rng_key, Y=Y, t=t)
        model_kwargs = dict(Y=Y, t=t)

    # Point-wise log-likelihood
    samples_grouped = mcmc.get_samples(group_by_chain=True)
    ll = log_likelihood(model_fn, samples_grouped, batch_ndims=2,
**model_kwargs)["Y_obs"]

    # InferenceData
    idata = az.from_numpyro(mcmc)
    idata.log_likelihood = xr.Dataset({"Y_obs":
(("chain","draw","obs_dim_0","obs_dim_1"), np.asarray(ll))})

    tag = "quadratic" if quadratic else "linear"
```

```
    az.to_netcdf(idata, os.path.join(DIR_CMP,
f"idata_table2_{var_label}_{tag}.nc"))

    # LOO / LOOIC (compatible with different arviz versions)
    loo = az.loo(idata, var_name="Y_obs", pointwise=False)
    def _get(obj, candidates, default=np.nan):
        for nm in candidates:
            if hasattr(obj, nm):
                return getattr(obj, nm)
        return default
    elpd_loo = _get(loo, ["elpd_loo","elpd"], np.nan)
    se_loo   = _get(loo, ["elpd_loo_se","se"], np.nan)
    looic    = _get(loo, ["looic","ic"], np.nan)

    # Approximate BIC (using posterior means; for comparison only, not
strict IC)
    def post_mean(name): return np.mean(idata.posterior[name].values,
axis=(0,1))
    a = post_mean("alpha"); b = post_mean("beta")
    mu = a[:,None] + b[:,None]*np.asarray(t)[None,:]
    if quadratic and "beta2" in idata.posterior:
        b2 = post_mean("beta2")
        mu = mu + b2[:,None]*(np.asarray(t)**2)[None,:]
    sd = float(post_mean("sigma_y"))
    y = np.asarray(Y)
    ll_tot = (-0.5*np.log(2*np.pi*sd*sd) - 0.5*((y-
mu)**2)/(sd*sd)).sum()

    k_params = sum(int(np.prod(v.values.shape[2:])) for v in
idata.posterior.data_vars.values())
    bic_like = -2.0*float(ll_tot) + k_params*np.log(y.size)

    # Cleanup
    gc.collect()
    try:
        jax.clear_caches()
    except Exception:
        pass

    return idata, dict(elpd_loo=elpd_loo, se=se_loo, looic=looic),
dict(bic_like=bic_like)

# %% 7) Loop through variables: linear/quadratic
rows_per_model, rows_diff = [], []

for vlabel, col in VAR_LIST:
    Y, t, t2, yrs, las = build_panel_for(col)
```

```
    # Linear
    _, loo_lin, bic_lin = run_one(vlabel, Y, t, t2, quadratic=False)
    # Quadratic
    _, loo_quad, bic_quad = run_one(vlabel, Y, t, t2, quadratic=True)

    rows_per_model += [
        {"Variable": vlabel, "Model":"Linear",
         "ELPD LOO": loo_lin["elpd_loo"], "S.E.": loo_lin["se"],
         "LOOIC": loo_lin["looic"], "BIC_like": bic_lin["bic_like"]},
        {"Variable": vlabel, "Model":"Quadratic",
         "ELPD LOO": loo_quad["elpd_loo"], "S.E.": loo_quad["se"],
         "LOOIC": loo_quad["looic"], "BIC_like": bic_quad["bic_like"]},
    ]

    # Difference (Linear - Quadratic)
    d_elpd = (loo_lin["elpd_loo"] - loo_quad["elpd_loo"]) if
(np.isfinite(loo_lin["elpd_loo"]) and
np.isfinite(loo_quad["elpd_loo"])) else np.nan
    d_se   = np.sqrt((loo_lin["se"] or np.nan)**2 + (loo_quad["se"] or
np.nan)**2) if (loo_lin["se"] is not None and loo_quad["se"] is not
None) else np.nan
    ratio  = (d_elpd / d_se) if (np.isfinite(d_elpd) and
np.isfinite(d_se) and d_se>0) else np.nan
    rows_diff.append({"Variable": vlabel,
                      "ELPD diff. (Linear - Quadratic)": d_elpd,
                      "S.E. diff.": d_se,
                      "ELPD/S.E. ratio": ratio})

# Export (CSV + Excel)
table_per_model = pd.DataFrame(rows_per_model)
table_diff     = pd.DataFrame(rows_diff)

csv1 = os.path.join(DIR_CMP, "model_compare_table2_by_variable.csv")
csv2 = os.path.join(DIR_CMP,
"model_compare_table2_by_variable_diff.csv")
xlsx = os.path.join(DIR_CMP, "model_compare_table2_by_variable.xlsx")

table_per_model.to_csv(csv1, index=False, encoding="utf-8-sig")
table_diff.to_csv(csv2, index=False, encoding="utf-8-sig")
with pd.ExcelWriter(xlsx) as w:
    table_per_model.to_excel(w, index=False, sheet_name="per_model")
    table_diff.to_excel(w, index=False, sheet_name="diff_by_var")

print("Saved:", os.path.join(DIR_PREP, "clean_panel_basic.csv"))
print("Saved:", os.path.join(DIR_PREP, "audit_classification.csv"))
print("Saved:", os.path.join(DIR_PREP, "group_counts.csv"))
print("Saved:", csv1)
print("Saved:", csv2)
```

```
print("Saved:", xlsx)
print("Done Part-1 (raw data, no TVC): variable-wise Linear vs
Quadratic.")
```

In [ ]:

```
# Part 2: Subsample parallel linear growth models (Bayesian / NumPyro)
# A: Long-term party groups (8 years unchanged: CON/LAB/NOC)
# B: Net switchers (first≠last year, only LAB->CON, LAB->NOC, CON->LAB,
CON->NOC, NOC->LAB, NOC->CON)
# Specification (main):
#   - Linear time term (t, centered on 2016..2023)
#   - Observation-level TVCs: Covid pulse(2020=1) + exponential
recovery rec_exp=1-exp(-rho*recovery)
#   - Mediation only at slope level (no within-level residual xi)
#   - FULL: Control variables (Mean, Unemp_Rate, P_Density) enter slope
level
#   - PARTY-ONLY: Only party/direction effects, no mediation or
controls
# Outputs (each model generates its own):
#   - Figure 1 (forest plot): Key parameters' 89% HDI (OVERALL; Full
also outputs LATE)
#   - Figure 2: Existing traj_* trajectory plots, not regenerated here
#   - Table 3 (OVERALL slope level) / Table 4 (LATE slope level; Full
only)

# %% 0) Imports & Config
import os, warnings, gc, random as pyrandom
from typing import Dict, List, Tuple
import numpy as np
import pandas as pd
import arviz as az
import xarray as xr
import matplotlib.pyplot as plt

import jax
import jax.numpy as jnp
from jax import random

import numpyro
import numpyro.distributions as dist
from numpyro.infer import MCMC, NUTS

# log_likelihood for different NumPyro versions
try:
    from numpyro.infer import log_likelihood
except Exception:
    from numpyro.infer.util import log_likelihood

warnings.filterwarnings("ignore")
```

```
numpyro.enable_x64()

# Matplotlib style
plt.rcParams["figure.figsize"] = (9, 5)
plt.rcParams["figure.dpi"] = 130
plt.rcParams["font.family"] = "Times New Roman"

# Paths & constants
DATA_PATH = r"F:/文档/UB/MGRCM0042/Data/Analysis/Full_Data_Final.xlsx"
OUT_ROOT  = r"F:/文档/UB/MGRCM0042/Data/Analysis/Result/Part2_MainSpecs"
os.makedirs(OUT_ROOT, exist_ok=True)

YEAR_START, YEAR_END = 2016, 2023
Y_MIN, Y_MAX = 2.0, 3.0  # Uniform y-axis range for trajectory plots
SEED = 42
CHAINS = 5
WARMUP = 3000
SAMPLES = 2000
TARGET_ACCEPT = 0.99

# Party code mapping: Plan_P ∈ {0,1,2} = {LAB, CON, NOC}
PARTY_MAP = {0: "LAB", 1: "CON", 2: "NOC"}
A_BASELINE_LABEL = "CON"  # A baseline & reference in plots
CON_DEEP_GRAY = "#3a3a3a"

# Color palette (for B plots)
PALETTE = [
    "#1f77b4", "#ff7f0e", "#2ca02c",
    "#d62728", "#9467bd", "#8c564b"
]

# %% 1) Load & base cleaning
df = pd.read_excel(DATA_PATH)
df.columns = [c.strip() for c in df.columns]
need =
["ONSCode","Year","Plan_P","Overall","LATE_P18","Mean","Unemp_Rate","P_
Density","Covid"]
miss = [c for c in need if c not in df.columns]
if miss:
    raise ValueError(f"Missing required columns: {miss}")

df = df[(df["Year"]>=YEAR_START) & (df["Year"]<=YEAR_END)].copy()
df = df.dropna(subset=["ONSCode","Year"]).copy()
df = df.sort_values(["ONSCode","Year"]).reset_index(drop=True)

# Cast numeric
```

```
for c in
["Plan_P","Overall","LATE_P18","Mean","Unemp_Rate","P_Density","Covid"]
:
    df[c] = pd.to_numeric(df[c], errors="coerce")

# Check continuous years
years = sorted(df["Year"].unique().tolist())
expected = list(range(YEAR_START, YEAR_END+1))
if years != expected:
    raise AssertionError(f"Years must be continuous {expected}, got
{years}")
T = len(years)

# Time variables (centered)
t_vals = np.array([y - YEAR_START for y in years], dtype=float)       #
0..7
t_c = t_vals - t_vals.mean()                                    #
centered
t = jnp.asarray(t_c)

# Covid TVCs
df["pulse"] = (df["Year"]==2020).astype(float)
df["recovery"] = np.clip(df["Year"]-2020, 0, None).astype(float)  #
2020=0; 2021..2023=1,2,3

# Plan_P → Party label
df["Party"] = df["Plan_P"].map(PARTY_MAP).astype("category")

# Keep only necessary columns
df =
df[["ONSCode","Year","Party","Overall","LATE_P18","Mean","Unemp_Rate","
P_Density","pulse","recovery"]].copy()

# %% 2) Z standardization (unified scaler across full sample for A/B
comparability)
scaler: Dict[str, Dict[str, float]] = {}
def zfit(s: pd.Series, name: str):
    mu, sd = s.mean(), s.std(ddof=0)
    sd = sd if sd>0 else 1.0
    scaler[name] = {"mean": float(mu), "std": float(sd)}
    return (s - mu)/sd

for c in ["Overall","LATE_P18","Mean","Unemp_Rate","P_Density"]:
    df[f"{c}_z"] = zfit(df[c], c)

# Save scaler
pd.DataFrame(scaler).T.to_csv(os.path.join(OUT_ROOT,
"zscaler_all.csv"), encoding="utf-8-sig")
```

```
# %% 3) A/B split (A: 8 years unchanged; B: net switchers=first≠last;
exclude reverters)
def split_groups(df_all: pd.DataFrame):
    nunique_by_la = df_all.groupby("ONSCode")["Party"].nunique()
    first_party   = df_all.groupby("ONSCode")["Party"].first()
    last_party    = df_all.groupby("ONSCode")["Party"].last()

    # A: 8 years unchanged
    ons_A = nunique_by_la[nunique_by_la == 1].index.tolist()
    df_A = df_all[df_all["ONSCode"].isin(ons_A)].copy()
    df_A["A_group"] =
df_A["ONSCode"].map(first_party.astype(str).to_dict())

    # B: net switchers (first != last), exclude reverters (first=last
but changed in between)
    mask_B = (nunique_by_la > 1) & (first_party != last_party)
    ons_B = nunique_by_la[mask_B].index.tolist()
    df_B = df_all[df_all["ONSCode"].isin(ons_B)].copy()
    fp = first_party.loc[ons_B].astype(str)
    lp = last_party.loc[ons_B].astype(str)
    direction_map = (fp.str.cat(lp, sep="->")).to_dict()
    df_B["Direction"] = df_B["ONSCode"].map(direction_map)

    return df_A, df_B

df_A, df_B = split_groups(df)

print("[Counts] A (no-switch):")
print(df_A.groupby("A_group")["ONSCode"].nunique())
print("\n[Counts] B (net-switchers only):")
print(df_B.groupby("Direction")["ONSCode"].nunique())

# %% 4) Panel construction function
def build_panel(df_sub: pd.DataFrame, group_col: str):
    years_sub = sorted(df_sub["Year"].unique().tolist())
    if years_sub != years:
        raise ValueError("Subsample years incomplete.")
    las = df_sub["ONSCode"].astype("category").cat.categories.tolist()
    N = len(las)

    def wide(col):
        w = (df_sub.pivot_table(index="ONSCode", columns="Year",
values=col, aggfunc="mean")
                  .reindex(index=las, columns=years))
        if w.isna().any().any():
            raise ValueError(f"{col} expansion still has missing values")
        return w.values.astype(float)
```

```
    Y_over    = wide("Overall_z")
    Y_late    = wide("LATE_P18_z")
    C_mean    = wide("Mean_z")
    C_unemp   = wide("Unemp_Rate_z")
    C_pdens   = wide("P_Density_z")
    pulse     = wide("pulse")
    recovery  = wide("recovery")

    group = df_sub.groupby("ONSCode")[group_col].first().astype(str)
    levels = sorted(group.unique().tolist())
    G =
group.astype("category").cat.reorder_categories(levels).cat.codes.value
s.astype(int)

    return {
        "ons": las,
        "Y_over_z": Y_over,
        "Y_late_z": Y_late,
        "C": np.stack([C_mean, C_unemp, C_pdens], axis=-1),  # (N,T,3)
        "pulse": pulse,
        "recovery": recovery,
        "G": G,
        "levels": levels
    }

A_inputs = build_panel(df_A, "A_group")
B_inputs = build_panel(df_B, "Direction") if len(df_B)>0 else None

# Baseline index (A fixed as CON)
A_levels = A_inputs["levels"]
if A_BASELINE_LABEL not in A_levels:
    raise ValueError(f"A group does not contain {A_BASELINE_LABEL},
actual levels = {A_levels}")
baseline_id_A = A_levels.index(A_BASELINE_LABEL)

# B levels (for identification only)
if B_inputs is not None:
    B_levels = B_inputs["levels"]
    b_counts =
pd.Series(B_inputs["G"]).value_counts().reindex(range(len(B_levels)),
fill_value=0)
    baseline_id_B = int(b_counts.idxmax())
else:
    B_levels, baseline_id_B = [], None

print(f"\n[A] groups: {A_levels} | baseline = {A_BASELINE_LABEL}")
if B_inputs is not None:
```

```
    print(f"[B] groups: {B_levels} | baseline (identification) =
{B_levels[baseline_id_B]}")
else:
    print("[B] no net-switchers; skip Model B.")

# %% 5) NumPyro models (party-only / full; exponential recovery)
def model_party_only(Y_over, C, G, t, pulse, recovery, baseline_id:
int, nG: int):
    N, T = Y_over.shape
    sigma_over = numpyro.sample("sigma_over", dist.HalfNormal(1.0))

    mu_a = numpyro.sample("mu_alpha_over", dist.Normal(0.,1.))
    sd_a = numpyro.sample("sd_alpha_over", dist.HalfNormal(1.))
    mu_b = numpyro.sample("mu_beta_over",  dist.Normal(0.,1.))
    sd_b = numpyro.sample("sd_beta_over",  dist.HalfNormal(1.))
    with numpyro.plate("la_over", N):
        z_a = numpyro.sample("z_alpha_over", dist.Normal(0.,1.))
        z_b = numpyro.sample("z_beta_over",  dist.Normal(0.,1.))
        alpha_over = numpyro.deterministic("alpha_over", mu_a +
z_a*sd_a)
        beta_over_base = numpyro.deterministic("beta_over_base", mu_b +
z_b*sd_b)

    theta = numpyro.sample("theta_group_over",
dist.Normal(0.,2.).expand([nG]).to_event(1))
    theta = theta.at[baseline_id].set(0.0)
    theta_G = jnp.take(theta, G, axis=-1)
    beta_over = numpyro.deterministic("beta_over", beta_over_base +
theta_G)

    # TVC: pulse + exponential recovery
    k_p = numpyro.sample("k_over_pulse",    dist.Normal(0., 0.5))
    k_r = numpyro.sample("k_over_recovery", dist.Normal(0., 0.5))
    rho_over = numpyro.sample("rho_over_recovery",
dist.HalfNormal(1.0))
    rec_over = 1.0 - jnp.exp(-rho_over * recovery)  # (N,T)

    mu = alpha_over[:,None] + beta_over[:,None]*t[None,:]
    mu = mu + k_p * pulse + k_r * rec_over
    numpyro.sample("Y_over", dist.Normal(mu, sigma_over), obs=Y_over)

def model_full(Y_over, Y_late, C, G, t, pulse, recovery, baseline_id:
int, nG: int):
    N, T = Y_late.shape
    assert C.shape == (N,T,3)

    sigma_over = numpyro.sample("sigma_over", dist.HalfNormal(1.))
    sigma_late = numpyro.sample("sigma_late", dist.HalfNormal(1.))
```

```
    # LATE
    mu_al = numpyro.sample("mu_alpha_late", dist.Normal(0.,1.))
    sd_al = numpyro.sample("sd_alpha_late", dist.HalfNormal(1.))
    mu_bl = numpyro.sample("mu_beta_late",  dist.Normal(0.,1.))
    sd_bl = numpyro.sample("sd_beta_late",  dist.HalfNormal(1.))
    with numpyro.plate("la_late", N):
        z_al = numpyro.sample("z_alpha_late", dist.Normal(0.,1.))
        z_bl = numpyro.sample("z_beta_late",  dist.Normal(0.,1.))
        alpha_late = numpyro.deterministic("alpha_late", mu_al +
z_al*sd_al)
        beta_late_base = numpyro.deterministic("beta_late_base", mu_bl +
z_bl*sd_bl)

    # OVERALL
    mu_ao = numpyro.sample("mu_alpha_over", dist.Normal(0.,1.))
    sd_ao = numpyro.sample("sd_alpha_over", dist.HalfNormal(1.))
    mu_bo = numpyro.sample("mu_beta_over",  dist.Normal(0.,1.))
    sd_bo = numpyro.sample("sd_beta_over",  dist.HalfNormal(1.))
    with numpyro.plate("la_over", N):
        z_ao = numpyro.sample("z_alpha_over", dist.Normal(0.,1.))
        z_bo = numpyro.sample("z_beta_over",  dist.Normal(0.,1.))
        alpha_over = numpyro.deterministic("alpha_over", mu_ao +
z_ao*sd_ao)
        beta_over_base = numpyro.deterministic("beta_over_base", mu_bo +
z_bo*sd_bo)

    # Group effects
    theta_late = numpyro.sample("theta_group_late",
dist.Normal(0.,2.).expand([nG]).to_event(1))
    theta_over = numpyro.sample("theta_group_over",
dist.Normal(0.,2.).expand([nG]).to_event(1))
    theta_late = theta_late.at[baseline_id].set(0.0)
    theta_over = theta_over.at[baseline_id].set(0.0)
    theta_late_G = jnp.take(theta_late, G, axis=-1)
    theta_over_G = jnp.take(theta_over, G, axis=-1)

    # Controls → slope
    eta_ctrl_late   = numpyro.sample("eta_ctrl_late",
dist.Normal(0.,0.5).expand([3]).to_event(1))
    delta_ctrl_over = numpyro.sample("delta_ctrl_over",
dist.Normal(0.,0.5).expand([3]).to_event(1))
    C_bar = C.mean(axis=1)            # (N,3)
    ctrl_late = jnp.dot(C_bar, eta_ctrl_late)     # (N,)
    ctrl_over = jnp.dot(C_bar, delta_ctrl_over)   # (N,)

    # Slope layer
```

```
    beta_late = numpyro.deterministic("beta_late", beta_late_base +
theta_late_G + ctrl_late)
    b_med = numpyro.sample("b_late_over", dist.Normal(0.,0.7))
    beta_over = numpyro.deterministic(
        "beta_over",
        beta_over_base + theta_over_G + ctrl_over + b_med*beta_late
    )

    # TVC: pulse + exponential recovery (separate rho for late/over)
    k_lp = numpyro.sample("k_late_pulse",    dist.Normal(0., 0.5))
    k_lr = numpyro.sample("k_late_recovery", dist.Normal(0., 0.5))
    k_op = numpyro.sample("k_over_pulse",    dist.Normal(0., 0.5))
    k_or = numpyro.sample("k_over_recovery", dist.Normal(0., 0.5))

    rho_late = numpyro.sample("rho_late_recovery",
dist.HalfNormal(1.0))
    rho_over = numpyro.sample("rho_over_recovery",
dist.HalfNormal(1.0))

    rec_late = 1.0 - jnp.exp(-rho_late * recovery)  # (N,T)
    rec_over = 1.0 - jnp.exp(-rho_over * recovery)  # (N,T)

    mu_late = alpha_late[:,None] + beta_late[:,None]*t[None,:]
    mu_late = mu_late + k_lp * pulse + k_lr * rec_late

    mu_over = alpha_over[:,None] + beta_over[:,None]*t[None,:]
    mu_over = mu_over + k_op * pulse + k_or * rec_over

    numpyro.sample("Y_late", dist.Normal(mu_late, sigma_late),
obs=Y_late)
    numpyro.sample("Y_over", dist.Normal(mu_over, sigma_over),
obs=Y_over)

# %% 6) Runner: includes log_likelihood (compute LOO/WAIC based on
Y_over)
def run_numpyro(model_fn, name: str, save_dir: str, var_name_for_ic:
str, **kwargs):
    os.makedirs(save_dir, exist_ok=True)
    rng_key = random.PRNGKey(SEED)
    kernel = NUTS(model_fn, target_accept_prob=TARGET_ACCEPT)
    mcmc = MCMC(kernel, num_warmup=WARMUP, num_samples=SAMPLES,
num_chains=CHAINS, progress_bar=True)
    mcmc.run(rng_key, **kwargs)

    extra = mcmc.get_extra_fields(group_by_chain=True)
    div = extra.get("diverging", None)
    if div is not None:
        rate = float(np.asarray(div).mean())
```

```
        print(f"[{name}] Divergences:
total={int(np.asarray(div).sum())}, rate={rate:.2%}")

    idata = az.from_numpyro(mcmc)

    # Attach log_likelihood (grouped, batch_ndims=2)
    samples_grouped = mcmc.get_samples(group_by_chain=True)
    ll_parts = log_likelihood(model_fn, samples_grouped, batch_ndims=2,
**kwargs)  # dict
    ll_vars = {}
    for k, arr in ll_parts.items():
        ll_vars[k] = (("chain","draw","obs_dim_0","obs_dim_1"),
np.asarray(arr))
    idata.log_likelihood = xr.Dataset(data_vars=ll_vars)

    # Save LOO/WAIC (based on var_name_for_ic)
    try:
        loo = az.loo(idata, var_name=var_name_for_ic)
        waic = az.waic(idata, var_name=var_name_for_ic)
        fit_row = {
            "ELPD LOO": float(getattr(loo, "elpd_loo", np.nan)),
            "S.E.": float(getattr(loo, "elpd_loo_se", np.nan)),
            "LOOIC": float(getattr(loo, "looic", np.nan)),
            "WAIC": float(getattr(waic, "waic", np.nan)),
            "WAIC S.E.": float(getattr(waic, "waic_se", np.nan)),
        }
    except Exception as e:
        print(f"[{name}] LOO/WAIC calculation failed: {e}")
        fit_row = {"ELPD LOO": np.nan, "S.E.": np.nan, "LOOIC": np.nan,
"WAIC": np.nan, "WAIC S.E.": np.nan}

    pd.DataFrame([fit_row]).to_csv(os.path.join(save_dir,
f"fit_{name}.csv"), index=False, encoding="utf-8-sig")
    return idata

# %% 7) Basic utilities: extraction, statistics, ROPE
def get_post(idata, name: str):
    if name not in idata.posterior:
        return None
    return idata.posterior[name].values  # (chain, draw, ...)

def flat(x): return np.asarray(x).reshape(-1)
def hdi89_vals(x):
    lo, hi = az.hdi(flat(x), hdi_prob=0.89)
    return float(lo), float(hi)
def pdirection_vals(x):
    s = flat(x)
    return float(max((s>0).mean(), (s<0).mean()))
```

```
def rope_stats(x, rope=(-0.1, 0.1)):
    s = flat(x)
    below = float((s < rope[0]).mean())
    within = float(((s >= rope[0]) & (s <= rope[1])).mean())
    above = float((s > rope[1]).mean())
    return below, within, above

# %% 8) Generate "Table 3/4" (each model generates its own)
def make_table3_overall(idata, levels: List[str], save_csv: str,
model_label: str, ctrl_names=("Mean","Unemp_Rate","P_Density")):
    rows = []
    # Group effects (OVERALL slope)
    th_over = get_post(idata, "theta_group_over")
    if th_over is not None:
        for j, lab in enumerate(levels):
            s = th_over[..., j]
            mean = float(np.mean(s)); lo,hi = hdi89_vals(s); pdv =
pdirection_vals(s)
            b,w,a = rope_stats(s)
            rows.append({"model": model_label, "param":
f"theta_over[{lab}]", "mean": mean, "hdi89_low": lo, "hdi89_high": hi,
"pd": pdv, "rope_below": b, "rope_within": w, "rope_above": a})
    # Mediation (full only)
    b_med = get_post(idata, "b_late_over")
    if b_med is not None:
        s = b_med; mean=float(np.mean(s)); lo,hi=hdi89_vals(s);
pdv=pdirection_vals(s); b,w,a=rope_stats(s)
        rows.append({"model": model_label, "param": "b_late_over",
"mean": mean, "hdi89_low": lo, "hdi89_high": hi, "pd": pdv,
"rope_below": b, "rope_within": w, "rope_above": a})
    # TVC & recovery parameters
    for p in
["k_over_pulse","k_over_recovery","rho_over_recovery","sigma_over"]:
        arr = get_post(idata, p)
        if arr is not None:
            s = arr; mean=float(np.mean(s)); lo,hi=hdi89_vals(s);
pdv=pdirection_vals(s); b,w,a=rope_stats(s)
            rows.append({"model": model_label, "param": p, "mean": mean,
"hdi89_low": lo, "hdi89_high": hi, "pd": pdv, "rope_below": b,
"rope_within": w, "rope_above": a})
    # Control variables (full only)
    eta = get_post(idata, "delta_ctrl_over")
    if eta is not None:
        for k, nm in enumerate(ctrl_names):
            if k < eta.shape[-1]:
                s = eta[..., k]; mean=float(np.mean(s));
lo,hi=hdi89_vals(s); pdv=pdirection_vals(s); b,w,a=rope_stats(s)
```

```
            rows.append({"model": model_label, "param":
f"delta_ctrl_over[{nm}]", "mean": mean, "hdi89_low": lo, "hdi89_high":
hi, "pd": pdv, "rope_below": b, "rope_within": w, "rope_above": a})
    pd.DataFrame(rows).to_csv(save_csv, index=False, encoding="utf-8-
sig")

def make_table4_late(idata, levels: List[str], save_csv: str,
model_label: str, ctrl_names=("Mean","Unemp_Rate","P_Density")):
    rows = []
    th_late = get_post(idata, "theta_group_late")
    if th_late is None:
        pd.DataFrame(rows,
columns=["model","param","mean","hdi89_low","hdi89_high","pd","rope_bel
ow","rope_within","rope_above"]).to_csv(save_csv, index=False,
encoding="utf-8-sig")
        return
    for j, lab in enumerate(levels):
        s = th_late[..., j]
        mean=float(np.mean(s)); lo,hi=hdi89_vals(s);
pdv=pdirection_vals(s); b,w,a=rope_stats(s)
        rows.append({"model": model_label, "param":
f"theta_late[{lab}]", "mean": mean, "hdi89_low": lo, "hdi89_high": hi,
"pd": pdv, "rope_below": b, "rope_within": w, "rope_above": a})
    for p in
["k_late_pulse","k_late_recovery","rho_late_recovery","sigma_late"]:
        arr = get_post(idata, p)
        if arr is not None:
            s = arr; mean=float(np.mean(s)); lo,hi=hdi89_vals(s);
pdv=pdirection_vals(s); b,w,a=rope_stats(s)
            rows.append({"model": model_label, "param": p, "mean": mean,
"hdi89_low": lo, "hdi89_high": hi, "pd": pdv, "rope_below": b,
"rope_within": w, "rope_above": a})
    eta = get_post(idata, "eta_ctrl_late")
    if eta is not None:
        for k, nm in enumerate(ctrl_names):
            if k < eta.shape[-1]:
                s = eta[..., k]; mean=float(np.mean(s));
lo,hi=hdi89_vals(s); pdv=pdirection_vals(s); b,w,a=rope_stats(s)
                rows.append({"model": model_label, "param":
f"eta_ctrl_late[{nm}]", "mean": mean, "hdi89_low": lo, "hdi89_high":
hi, "pd": pdv, "rope_below": b, "rope_within": w, "rope_above": a})
    pd.DataFrame(rows).to_csv(save_csv, index=False, encoding="utf-8-
sig")

# %% 9) Trajectories & Figure 2: this script only generates traj_*
(Figure 2 exists, no re-generation of fig2_*)
```

```
def posterior_group_traj(idata, ons_list: List[str], df_sub:
pd.DataFrame, group_col: str,
                         levels: List[str], years: List[int], scaler:
Dict[str,Dict[str,float]],
                         alpha_name="alpha_over", beta_name="beta_over") ->
Dict[str, Tuple[np.ndarray, np.ndarray]]:
    a = get_post(idata, alpha_name)  # (chain,draw,N)
    b = get_post(idata, beta_name)   # (chain,draw,N)
    if a is None or b is None: return {}

    CD = a.shape[0]*a.shape[1]
    a = a.reshape(CD, a.shape[2])
    b = b.reshape(CD, b.shape[2])
    t_np = np.asarray(t)

    first_map =
df_sub.groupby("ONSCode")[group_col].first().astype(str).to_dict()
    groups = {lab: np.where([first_map[ons]==lab for ons in
ons_list])[0] for lab in levels}

    mu_over, sd_over = scaler["Overall"]["mean"],
scaler["Overall"]["std"]
    out = {}
    for lab, idx in groups.items():
        if len(idx)==0:
            continue
        a_g = a[:, idx].mean(axis=1)
        b_g = b[:, idx].mean(axis=1)
        traj_z = a_g[:,None] + b_g[:,None]*t_np[None,:]
        traj_raw = traj_z*sd_over + mu_over
        med = np.median(traj_raw, axis=0)
        h = az.hdi(traj_raw, hdi_prob=0.89)
        out[lab] = (med, h, traj_raw)
    return out

def plot_trajectories_with_ref(idata, ons_list, df_sub, group_col,
levels, years, scaler,
                               title, save_path, ref_label=None,
ref_median=None, ref_hdi=None,
                               baseline_label_for_color=None):
    traj = posterior_group_traj(idata, ons_list, df_sub, group_col,
levels, years, scaler)
    if not traj:
        print("No trajectories to plot."); return

    plt.figure()
    colors = {}
    if baseline_label_for_color and baseline_label_for_color in levels:
```

```
        colors[baseline_label_for_color] = CON_DEEP_GRAY
    others = [lab for lab in levels if lab != baseline_label_for_color]
    for i, lab in enumerate(others):
        colors[lab] = PALETTE[i % len(PALETTE)]

    for lab in levels:
        if lab not in traj:
            continue
        med, h, _ = traj[lab]
        c = colors.get(lab, "#444444")
        plt.plot(years, med, lw=2.2, label=lab, color=c)
        plt.fill_between(years, h[:,0], h[:,1], alpha=0.18, color=c)

    if (ref_label is not None) and (ref_median is not None) and
(ref_hdi is not None):
        plt.plot(years, ref_median, lw=2.8, color=CON_DEEP_GRAY,
label=ref_label, linestyle="-")
        plt.fill_between(years, ref_hdi[:,0], ref_hdi[:,1],
color=CON_DEEP_GRAY, alpha=0.12)

    plt.title(title)
    plt.xlabel("Year")
    plt.ylabel("OVERALL (raw)")
    plt.ylim(Y_MIN, Y_MAX)
    plt.legend(frameon=False, ncol=2)
    plt.tight_layout()
    plt.savefig(save_path, dpi=180, bbox_inches="tight")
    plt.close()

# %% 10) Forest plot utility (replaces original histogram fig1)
def _forest_plot(df_plot: pd.DataFrame, title: str, save_path: str,
unit_label: str="z"):
    """df_plot needs columns: label, mean, hdi_low, hdi_high (optional
pd, rope_*)."""
    if df_plot is None or df_plot.empty:
        print(f"[forest] {title} — No parameters to plot, skipping.");
return
    dfp = df_plot.copy()
    # Display top to bottom: by block order and original input order
    dfp["y"] = np.arange(len(dfp))[::-1]
    plt.figure(figsize=(10, max(2.5, 0.5*len(dfp)+1.2)))
    # Error bars
    plt.hlines(dfp["y"], dfp["hdi_low"], dfp["hdi_high"], lw=3,
color="#444444")
    # Point estimates
    plt.plot(dfp["mean"], dfp["y"], "o", ms=5, color="#111111")
    # Zero line
    plt.axvline(0, lw=1.2, ls="--", color="#666666")
```

```
    # Axes/title
    plt.yticks(dfp["y"], dfp["label"])
    plt.xlabel(f"Effect ({unit_label})")
    plt.title(title)
    plt.tight_layout()
    plt.savefig(save_path, dpi=220, bbox_inches="tight")
    plt.close()

def _arr_to_rows(arr, labels, omit_idx=None, prefix=None):
    """Convert posterior array to rows list by labels."""
    rows = []
    if arr is None: return rows
    L = len(labels)
    for j in range(L):
        if (omit_idx is not None) and (j == omit_idx):
            continue
        s = arr[..., j]
        mean = float(np.mean(s)); lo,hi = hdi89_vals(s)
        lab = f"{prefix}: {labels[j]}" if prefix else labels[j]
        rows.append({"label": lab, "mean": mean, "hdi_low": lo,
"hdi_high": hi})
    return rows

def build_forest_overall(idata, levels, baseline_idx,
ctrl_names=("Mean","Unemp_Rate","P_Density")):
    rows = []
    # Group effects (OVERALL slope)
    rows += _arr_to_rows(get_post(idata,"theta_group_over"), levels,
omit_idx=baseline_idx, prefix="Group(OVERALL)")
    # Mediation (if present)
    b = get_post(idata,"b_late_over")
    if b is not None:
        mean=float(np.mean(b)); lo,hi=hdi89_vals(b)
        rows.append({"label":"Mediator b(LATE→OVERALL)", "mean":mean,
"hdi_low":lo, "hdi_high":hi})
    # Controls (if present)
    delta = get_post(idata,"delta_ctrl_over")
    if delta is not None:
        for k, nm in enumerate(ctrl_names):
            s = delta[...,k]; mean=float(np.mean(s)); lo,hi=hdi89_vals(s)
            rows.append({"label":f"Control[{nm}] → OVERALL slope",
"mean":mean, "hdi_low":lo, "hdi_high":hi})
    # TVC + ρ
    for nm, lab in [("k_over_pulse","Pulse(2020) amplitude"),
                    ("k_over_recovery","Recovery amplitude k"),
                    ("rho_over_recovery","Recovery rate ρ")]:
        s = get_post(idata, nm)
        if s is not None:
```

```
            mean=float(np.mean(s)); lo,hi=hdi89_vals(s)
            rows.append({"label":lab, "mean":mean, "hdi_low":lo,
"hdi_high":hi})
    return pd.DataFrame(rows)

def build_forest_late(idata, levels, baseline_idx,
ctrl_names=("Mean","Unemp_Rate","P_Density")):
    rows = []
    # Group effects (LATE slope)
    rows += _arr_to_rows(get_post(idata,"theta_group_late"), levels,
omit_idx=baseline_idx, prefix="Group(LATE)")
    # Controls
    eta = get_post(idata,"eta_ctrl_late")
    if eta is not None:
        for k, nm in enumerate(ctrl_names):
            s = eta[...,k]; mean=float(np.mean(s)); lo,hi=hdi89_vals(s)
            rows.append({"label":f"Control[{nm}] → LATE slope",
"mean":mean, "hdi_low":lo, "hdi_high":hi})
    # TVC + ρ
    for nm, lab in [("k_late_pulse","Pulse(2020) amplitude (LATE)"),
                    ("k_late_recovery","Recovery amplitude k (LATE)"),
                    ("rho_late_recovery","Recovery rate ρ (LATE)")]:
        s = get_post(idata, nm)
        if s is not None:
            mean=float(np.mean(s)); lo,hi=hdi89_vals(s)
            rows.append({"label":lab, "mean":mean, "hdi_low":lo,
"hdi_high":hi})
    return pd.DataFrame(rows)

# %% 11) Run A first: party-only & full (baseline=CON; generate A:CON
reference trajectory)
DIR_A = os.path.join(OUT_ROOT, "A_longrun"); os.makedirs(DIR_A,
exist_ok=True)
print(f"\n[A] N={len(A_inputs['ons'])} LAs")

# Party-only
idata_A_party = run_numpyro(
    model_party_only, name="A_partyonly", save_dir=DIR_A,
var_name_for_ic="Y_over",
    Y_over=jnp.asarray(A_inputs["Y_over_z"]),
    C=None,
    G=jnp.asarray(A_inputs["G"]),
    t=jnp.asarray(t_c),
    pulse=jnp.asarray(A_inputs["pulse"]),
    recovery=jnp.asarray(A_inputs["recovery"]),
    baseline_id=baseline_id_A,
    nG=len(A_levels)
)
```

```
# Tables 3/4
make_table3_overall(idata_A_party, A_levels, os.path.join(DIR_A,
"table3_A_partyonly.csv"), "A_partyonly")
make_table4_late(  idata_A_party, A_levels, os.path.join(DIR_A,
"table4_A_partyonly.csv"), "A_partyonly")  # Empty table

# Figure 1 (forest plot: OVERALL)
forest_over_A_party = build_forest_overall(idata_A_party, A_levels,
baseline_id_A)
_forest_plot(forest_over_A_party, "Key effects on OVERALL (z; 89% HDI)
— A: Party-only",
           os.path.join(DIR_A, "fig1_overall_A_partyonly.png"),
unit_label="z")

# Full model
idata_A_full = run_numpyro(
    model_full, name="A_full", save_dir=DIR_A,
var_name_for_ic="Y_over",
    Y_over=jnp.asarray(A_inputs["Y_over_z"]),
    Y_late=jnp.asarray(A_inputs["Y_late_z"]),
    C=jnp.asarray(A_inputs["C"]),
    G=jnp.asarray(A_inputs["G"]),
    t=jnp.asarray(t_c),
    pulse=jnp.asarray(A_inputs["pulse"]),
    recovery=jnp.asarray(A_inputs["recovery"]),
    baseline_id=baseline_id_A,
    nG=len(A_levels)
)
make_table3_overall(idata_A_full, A_levels, os.path.join(DIR_A,
"table3_A_full.csv"), "A_full")
make_table4_late(  idata_A_full, A_levels, os.path.join(DIR_A,
"table4_A_full.csv"), "A_full")

# Figure 1 (forest plots: OVERALL + LATE)
forest_over_A_full = build_forest_overall(idata_A_full, A_levels,
baseline_id_A)
_forest_plot(forest_over_A_full, "Key effects on OVERALL (z; 89% HDI) —
A: Full",
           os.path.join(DIR_A, "fig1_overall_A_full.png"),
unit_label="z")
forest_late_A_full = build_forest_late(idata_A_full, A_levels,
baseline_id_A)
_forest_plot(forest_late_A_full, "Key effects on LATE (z; 89% HDI) — A:
Full",
           os.path.join(DIR_A, "fig1_late_A_full.png"), unit_label="z")

# Trajectories (A: display CON in deep gray; for B overlay reference)
def _get_A_CON_ref(idata):
```

```
    traj = posterior_group_traj(idata, A_inputs["ons"], df_A,
"A_group", A_levels, years, scaler)
    return traj.get("CON", (None, None, None))[:2]

plot_trajectories_with_ref(
    idata_A_party, A_inputs["ons"], df_A, "A_group", A_levels, years,
scaler,
    title="Model A (party-only): OVERALL (raw) by long-run party",
    save_path=os.path.join(DIR_A, "traj_A_partyonly_OVERALL.png"),
    ref_label=None, ref_median=None, ref_hdi=None,
    baseline_label_for_color=A_BASELINE_LABEL
)
plot_trajectories_with_ref(
    idata_A_full, A_inputs["ons"], df_A, "A_group", A_levels, years,
scaler,
    title="Model A (full): OVERALL (raw) by long-run party",
    save_path=os.path.join(DIR_A, "traj_A_full_OVERALL.png"),
    ref_label=None, ref_median=None, ref_hdi=None,
    baseline_label_for_color=A_BASELINE_LABEL
)

# A:CON reference trajectory for B overlay
A_con_traj_party = _get_A_CON_ref(idata_A_party)
A_con_traj_full  = _get_A_CON_ref(idata_A_full)
if (A_con_traj_party[0] is None) or (A_con_traj_full[0] is None):
    print("Warning: A:CON reference trajectory unavailable, B plots
will not overlay reference.")

# %% 12) Run B (if exists): net switchers (party-only & full), overlay
A:CON reference; forest plots & tables 3/4
if B_inputs is not None and len(B_inputs["ons"])>0:
    DIR_B = os.path.join(OUT_ROOT, "B_switchers"); os.makedirs(DIR_B,
exist_ok=True)
    print(f"\n[B] N={len(B_inputs['ons'])} LAs")

    # Party-only
    idata_B_party = run_numpyro(
        model_party_only, name="B_partyonly", save_dir=DIR_B,
var_name_for_ic="Y_over",
        Y_over=jnp.asarray(B_inputs["Y_over_z"]),
        C=None,
        G=jnp.asarray(B_inputs["G"]),
        t=jnp.asarray(t_c),
        pulse=jnp.asarray(B_inputs["pulse"]),
        recovery=jnp.asarray(B_inputs["recovery"]),
        baseline_id=baseline_id_B,
        nG=len(B_levels)
    )
```

```
    make_table3_overall(idata_B_party, B_levels, os.path.join(DIR_B,
"table3_B_partyonly.csv"), "B_partyonly")
    make_table4_late(  idata_B_party, B_levels, os.path.join(DIR_B,
"table4_B_partyonly.csv"), "B_partyonly")  # Empty table

    # Forest plot (OVERALL)
    forest_over_B_party = build_forest_overall(idata_B_party, B_levels,
baseline_id_B)
    _forest_plot(forest_over_B_party, "Key effects on OVERALL (z; 89%
HDI) — B: Party-only",
                 os.path.join(DIR_B, "fig1_overall_B_partyonly.png"),
unit_label="z")

    # Full model
    idata_B_full = run_numpyro(
        model_full, name="B_full", save_dir=DIR_B,
var_name_for_ic="Y_over",
        Y_over=jnp.asarray(B_inputs["Y_over_z"]),
        Y_late=jnp.asarray(B_inputs["Y_late_z"]),
        C=jnp.asarray(B_inputs["C"]),
        G=jnp.asarray(B_inputs["G"]),
        t=jnp.asarray(t_c),
        pulse=jnp.asarray(B_inputs["pulse"]),
        recovery=jnp.asarray(B_inputs["recovery"]),
        baseline_id=baseline_id_B,
        nG=len(B_levels)
    )
    make_table3_overall(idata_B_full, B_levels, os.path.join(DIR_B,
"table3_B_full.csv"), "B_full")
    make_table4_late(  idata_B_full, B_levels, os.path.join(DIR_B,
"table4_B_full.csv"), "B_full")

    # Forest plots (OVERALL + LATE)
    forest_over_B_full = build_forest_overall(idata_B_full, B_levels,
baseline_id_B)
    _forest_plot(forest_over_B_full, "Key effects on OVERALL (z; 89%
HDI) — B: Full",
                 os.path.join(DIR_B, "fig1_overall_B_full.png"),
unit_label="z")
    forest_late_B_full = build_forest_late(idata_B_full, B_levels,
baseline_id_B)
    _forest_plot(forest_late_B_full, "Key effects on LATE (z; 89% HDI)
— B: Full",
                 os.path.join(DIR_B, "fig1_late_B_full.png"),
unit_label="z")

    # Trajectories with A:CON reference overlay
    ref_med_party, ref_hdi_party = A_con_traj_party
```

```
    plot_trajectories_with_ref(
        idata_B_party, B_inputs["ons"], df_B, "Direction", B_levels,
years, scaler,
        title="Model B (party-only): OVERALL (raw) by net-switch
direction (+ A:CON ref)",
        save_path=os.path.join(DIR_B, "traj_B_partyonly_OVERALL.png"),
        ref_label=("A: CON (reference)" if ref_med_party is not None
else None),
        ref_median=ref_med_party, ref_hdi=ref_hdi_party,
        baseline_label_for_color=None
    )
    ref_med_full, ref_hdi_full = A_con_traj_full
    plot_trajectories_with_ref(
        idata_B_full, B_inputs["ons"], df_B, "Direction", B_levels,
years, scaler,
        title="Model B (full): OVERALL (raw) by net-switch direction (+
A:CON ref)",
        save_path=os.path.join(DIR_B, "traj_B_full_OVERALL.png"),
        ref_label=("A: CON (reference)" if ref_med_full is not None else
None),
        ref_median=ref_med_full, ref_hdi=ref_hdi_full,
        baseline_label_for_color=None
    )

    # Comparison table (A_full vs B_full)
    def load_tbl(p):
        return pd.read_csv(p) if os.path.exists(p) else pd.DataFrame()
    cmp_rows = []
    tblA = load_tbl(os.path.join(DIR_A, "table3_A_full.csv"))
    tblB = load_tbl(os.path.join(DIR_B, "table3_B_full.csv"))
    if not tblA.empty:
        tblA["model"] = "A_full"; cmp_rows.append(tblA)
    if not tblB.empty:
        tblB["model"] = "B_full"; cmp_rows.append(tblB)
    if cmp_rows:
        cmp = pd.concat(cmp_rows, ignore_index=True)
        cmp.to_csv(os.path.join(OUT_ROOT,
"table3_compare_AvsB_full.csv"), index=False, encoding="utf-8-sig")
else:
    print("\n[B] No net-switching LAs (reverters excluded), skipping
Model B.")

gc.collect()
print("\n== Part 2 completed: Check output directory:", OUT_ROOT)
print("Generated files (examples):")
print("  A_longrun/: fig1_overall_A_partyonly.png,
fig1_overall_A_full.png, fig1_late_A_full.png, traj_A_*.png,
table3_*.csv, table4_*.csv")
```

```
print("  B_switchers/: fig1_overall_B_partyonly.png,
fig1_overall_B_full.png, fig1_late_B_full.png, traj_B_*.png,
table3_*.csv, table4_*.csv")
```

In [ ]:

```
# Part 3: Robustness (R) — Joint model on full sample (exclude
"reverters")
# 1) Group-effect priors: Normal(0, 2.0)
# 2) Mediator→outcome slope prior: Normal(0, 0.7)
# 3) Standardization: use shared scaler saved by Code 1
(zscaler_all.csv)
# 4) B-group baseline: most frequent transition as baseline
# 5) Covid: pulse = 1{Year==2020}; recovery = max(Year-2020, 0)
# 6) Naming aligned to Code 1 (k_over_*, k_late_*, eta_ctrl_late,
delta_ctrl_over)
# Other settings remain consistent with Code 1 (sampling, exponential
recovery, centered time).

# %% 0) Imports & config
import os, warnings, gc
from typing import Dict, List, Tuple
import numpy as np
import pandas as pd
import arviz as az
import matplotlib.pyplot as plt
import xarray as xr

import jax
import jax.numpy as jnp
from jax import random

import numpyro
import numpyro.distributions as dist
from numpyro.infer import MCMC, NUTS
try:
    from numpyro.infer import log_likelihood
except ImportError:
    from numpyro.infer.util import log_likelihood

warnings.filterwarnings("ignore")
numpyro.enable_x64()

plt.rcParams["figure.figsize"] = (9,5)
plt.rcParams["figure.dpi"] = 140
plt.rcParams["font.family"] = "Times New Roman"

DATA_PATH = r"F:/文档/UB/MGRCM0042/Data/Analysis/Full_Data_Final.xlsx"
OUT_ROOT  = r"F:/文档
/UB/MGRCM0042/Data/Analysis/Result/robust_joint_excl_reverters_aligned"
```

```
os.makedirs(OUT_ROOT, exist_ok=True)

# Load shared z-scaler produced by Code 1
SCALER_CSV = r"F:/文档
/UB/MGRCM0042/Data/Analysis/Result/Part2_MainSpecs/zscaler_all.csv"
if not os.path.exists(SCALER_CSV):
    raise FileNotFoundError(f"No scaler file: {SCALER_CSV}.")
scaler_df = pd.read_csv(SCALER_CSV, index_col=0)
scaler: Dict[str, Dict[str, float]] = scaler_df.to_dict(orient="index")

SEED = 42
CHAINS = 5
WARMUP = 3000
SAMPLES = 2000
TARGET_ACCEPT = 0.99

YEAR_START, YEAR_END = 2016, 2023
YEARS = list(range(YEAR_START, YEAR_END+1))
T = len(YEARS)
YLIM = (2.0, 3.5)

PARTY_MAP = {0:"LAB", 1:"CON", 2:"NOC"}  # from Plan_P

# Colors (match Part 2)
CON_DEEP_GRAY = "#3a3a3a"
PALETTE = [
    "#1f77b4", "#ff7f0e", "#2ca02c",
    "#d62728", "#9467bd", "#8c564b"
]

# ==========================================================
# 1) Load & basic clean
# ==========================================================
df_raw = pd.read_excel(DATA_PATH)
df_raw.columns = [c.strip() for c in df_raw.columns]

req =
["ONSCode","Year","Plan_P","Overall","Mean","Unemp_Rate","P_Density","L
ATE_P18","Covid"]
miss = [c for c in req if c not in df_raw.columns]
if miss:
    raise ValueError(f"Missing required columns: {miss}")

df = df_raw.loc[df_raw["Year"].between(YEAR_START, YEAR_END)].copy()
df = df.dropna(subset=["ONSCode","Year"]).copy()
for c in req:
    if c!="ONSCode":
        df[c] = pd.to_numeric(df[c], errors="coerce")
```

```
yrs = sorted(df["Year"].unique().tolist())
if yrs != YEARS:
    raise AssertionError(f"Years must be consecutive {YEARS}, actual
{yrs}")
cnt = df.groupby("ONSCode")["Year"].nunique()
bad = cnt[cnt!=T]
if not bad.empty:
    raise AssertionError(f"There is non-equilibrium LA:
{bad.to_dict()}")

# ==========================================================
# 2) Exclude "reverters": changed at some point but first==last
# ==========================================================
first_party = (df.sort_values(["ONSCode","Year"])\
              .groupby("ONSCode")["Plan_P"].first()).map(PARTY_MAP)
last_party  = (df.sort_values(["ONSCode","Year"])\
              .groupby("ONSCode")["Plan_P"].last()).map(PARTY_MAP)
nuniq = df.groupby("ONSCode")["Plan_P"].nunique()

reverters_idx = nuniq[(nuniq>1) & (first_party==last_party)].index
print(f"[Info] Excluding reverters (changed but first==last):
n={len(reverters_idx)}")

df = df[~df["ONSCode"].isin(reverters_idx)].copy()

# Recompute after exclusion
first_party = (df.sort_values(["ONSCode","Year"])\
              .groupby("ONSCode")["Plan_P"].first()).map(PARTY_MAP)
last_party  = (df.sort_values(["ONSCode","Year"])\
              .groupby("ONSCode")["Plan_P"].last()).map(PARTY_MAP)
nuniq = df.groupby("ONSCode")["Plan_P"].nunique()

# A: nunique==1; B: first!=last (net-switchers)
A_la = nuniq[nuniq==1].index
B_la = nuniq[(nuniq>1) & (first_party!=last_party)].index

df["A_or_B"] = np.where(df["ONSCode"].isin(A_la), "A",
np.where(df["ONSCode"].isin(B_la), "B", "EXCLUDE"))
df = df[df["A_or_B"].isin(["A","B"])].copy()

A_group = first_party.loc[A_la]
B_dir   = (first_party.loc[B_la].astype(str) + "->" +
last_party.loc[B_la].astype(str))

print("\n[Counts] A (no-switch):")
print(A_group.value_counts())
```

```
print("\n[Counts] B (net-switchers only):")
print(B_dir.value_counts())

# Group levels
A_levels = ["CON","LAB","NOC"]  # A baseline fixed as CON
B_levels =
["CON->LAB","CON->NOC","LAB->CON","LAB->NOC","NOC->CON","NOC->LAB"]
B_levels = [g for g in B_levels if g in B_dir.unique().tolist()]

# Labels for plotting
A_label_map = A_group.to_dict()
B_label_map = B_dir.to_dict()
df["A_group"] = df["ONSCode"].map(A_label_map)
df["Direction"] = df["ONSCode"].map(B_label_map)

# ==========================================================
# 3) Standardize using shared scaler from Code 1
# ==========================================================
def z_using_shared_scaler(series: pd.Series, key: str) -> pd.Series:
    if key not in scaler:
        raise KeyError(f"scaler missing {key} 's mean/std")
    mu = float(scaler[key]["mean"]) ; sd = float(scaler[key]["std"]) or
1.0
    return (series - mu) / (sd if sd>0 else 1.0)

for c in ["Overall","LATE_P18","Mean","Unemp_Rate","P_Density"]:
    df[f"{c}_z"] = z_using_shared_scaler(df[c], c)

# Time (centered)
df["t"] = df["Year"] - YEAR_START
t_vals = np.array(sorted(df["t"].unique()), dtype=float)
t_c = t_vals - t_vals.mean()
t = jnp.asarray(t_c)

# Covid TVCs aligned to Code 1: pulse = 1{Year==2020}; recovery =
max(Year-2020, 0)
years_arr = np.array(YEARS)
pulse_vec = (years_arr == 2020).astype(float)
recovery_vec = np.clip(years_arr - 2020, 0, None).astype(float)  #
2021=1, 2022=2, 2023=3

# ==========================================================
# 4) Build balanced panel arrays (after exclusion)
# ==========================================================
def to_wide(df_in: pd.DataFrame, value_col: str) -> Tuple[np.ndarray,
List[str]]:
    cats = df_in["ONSCode"].astype("category")
    las = cats.cat.categories.tolist()
```

```
    wide = (df_in.pivot_table(index="ONSCode", columns="Year",
values=value_col, aggfunc="mean")
            .reindex(index=las, columns=YEARS))
    if wide.isna().any().any():
        raise ValueError(f"{value_col} still missing after expansion")
    return wide.values.astype(float), las

Y_over_z, ons_list = to_wide(df, "Overall_z")
Y_late_z, _        = to_wide(df, "LATE_P18_z")
N = len(ons_list)

pulse = np.tile(pulse_vec, (N,1)).astype(float)
recovery = np.tile(recovery_vec, (N,1)).astype(float)

# Time-invariant controls per LA
C_mean  =
df.groupby("ONSCode")["Mean_z"].mean().reindex(ons_list).values.astype(
float)
C_unemp =
df.groupby("ONSCode")["Unemp_Rate_z"].mean().reindex(ons_list).values.a
stype(float)
C_pdens =
df.groupby("ONSCode")["P_Density_z"].mean().reindex(ons_list).values.as
type(float)
C = np.stack([C_mean, C_unemp, C_pdens], axis=1)  # (N, 3)

# One-hot blocks for A and B
A_idx = pd.Categorical(pd.Series([A_label_map.get(la, None) for la in
ons_list]), categories=A_levels)
A_ids = A_idx.codes
B_idx = pd.Categorical(pd.Series([B_label_map.get(la, None) for la in
ons_list]), categories=B_levels)
B_ids = B_idx.codes

nA = len(A_levels)
nB = len(B_levels)
GA = np.zeros((N, nA), dtype=float)
GB = np.zeros((N, nB), dtype=float)
for i, (aid, bid) in enumerate(zip(A_ids, B_ids)):
    if aid >= 0:
        GA[i, aid] = 1.0
    if bid >= 0:
        GB[i, bid] = 1.0

# B baseline = most frequent transition (align Code 1)
if nB > 0:
    b_code_series = pd.Series(B_ids)
    b_code_series = b_code_series[b_code_series >= 0]
```

```
    if b_code_series.empty:
        baseline_id_B = 0
    else:
        baseline_id_B = int(b_code_series.value_counts().idxmax())
else:
    baseline_id_B = 0

# Save audit
pd.DataFrame({"ONSCode":ons_list,
              "A_group":A_idx.astype(object),
              "Direction":B_idx.astype(object)})\
  .to_csv(os.path.join(OUT_ROOT,
"audit_R_groups_after_excluding_reverters.csv"), index=False,
encoding="utf-8-sig")

# ==========================================================
# 5) Model — Joint (exponential recovery; priors/names aligned)
# ==========================================================
def model_R_full(
    Y_over, Y_late, C, GA, GB, t, pulse, recovery,
    baseline_id_A: int, baseline_id_B: int
):
    """
    Structure:
      beta_late_i = beta_late_base_i + GA_i·thetaA_late +
GB_i·thetaB_late + C_i·eta_ctrl_late
      beta_over_i = beta_over_base_i + b_late_over * beta_late_i +
GA_i·phiA_over + GB_i·phiB_over + C_i·delta_ctrl_over
      Observation: alpha + beta * t + k_pulse * 1{2020} + k_rec * (1 -
exp(-rho * recovery))
    Priors aligned with Code 1:
      - Group effects ~ Normal(0, 2)
      - b_late_over ~ Normal(0, 0.7)
      - Controls on slopes ~ Normal(0, 0.5)
      - Recovery rates ~ HalfNormal(1)
    Baselines:
      - A baseline fixed at "CON"
      - B baseline = most frequent transition
    """
    N, T = Y_over.shape
    nA = GA.shape[1]; nB = GB.shape[1]
    Cdim = C.shape[1]

    # Noise
    sigma_over = numpyro.sample("sigma_over", dist.HalfNormal(1.))
    sigma_late = numpyro.sample("sigma_late", dist.HalfNormal(1.))

    # Per-LA intercepts/slopes (non-centered)
```

```
    mu_a_o = numpyro.sample("mu_alpha_over", dist.Normal(0.,1.))
    sd_a_o = numpyro.sample("sd_alpha_over", dist.HalfNormal(1.))
    mu_b_o = numpyro.sample("mu_beta_over",  dist.Normal(0.,1.))
    sd_b_o = numpyro.sample("sd_beta_over",  dist.HalfNormal(1.))

    mu_a_l = numpyro.sample("mu_alpha_late", dist.Normal(0.,1.))
    sd_a_l = numpyro.sample("sd_alpha_late", dist.HalfNormal(1.))
    mu_b_l = numpyro.sample("mu_beta_late",  dist.Normal(0.,1.))
    sd_b_l = numpyro.sample("sd_beta_late",  dist.HalfNormal(1.))

    with numpyro.plate("la_over", N):
        z_ao = numpyro.sample("z_alpha_over", dist.Normal(0.,1.))
        z_bo = numpyro.sample("z_beta_over",  dist.Normal(0.,1.))
        alpha_over = numpyro.deterministic("alpha_over", mu_a_o +
z_ao*sd_a_o)
        beta_over_base = numpyro.deterministic("beta_over_base", mu_b_o
+ z_bo*sd_b_o)

    with numpyro.plate("la_late", N):
        z_al = numpyro.sample("z_alpha_late", dist.Normal(0.,1.))
        z_bl = numpyro.sample("z_beta_late",  dist.Normal(0.,1.))
        alpha_late = numpyro.deterministic("alpha_late", mu_a_l +
z_al*sd_a_l)
        beta_late_base = numpyro.deterministic("beta_late_base", mu_b_l
+ z_bl*sd_b_l)

    # Group effects (A/B) — Normal(0, 2)
    thetaA_late = numpyro.sample("thetaA_late",
dist.Normal(0.,2.).expand([nA]).to_event(1))
    phiA_over   = numpyro.sample("phiA_over",
dist.Normal(0.,2.).expand([nA]).to_event(1))
    thetaA_late = thetaA_late.at[baseline_id_A].set(0.0)
    phiA_over   = phiA_over.at[baseline_id_A].set(0.0)

    thetaB_late = numpyro.sample("thetaB_late",
dist.Normal(0.,2.).expand([nB]).to_event(1)) if nB>0 else
jnp.zeros((0,))
    phiB_over   = numpyro.sample("phiB_over",
dist.Normal(0.,2.).expand([nB]).to_event(1)) if nB>0 else
jnp.zeros((0,))
    if nB>0:
        thetaB_late = thetaB_late.at[baseline_id_B].set(0.0)
        phiB_over   = phiB_over.at[baseline_id_B].set(0.0)

    # Controls on slopes
    eta_ctrl_late   = numpyro.sample("eta_ctrl_late",
dist.Normal(0.,0.5).expand([Cdim]).to_event(1))
```

```
    delta_ctrl_over = numpyro.sample("delta_ctrl_over",
dist.Normal(0.,0.5).expand([Cdim]).to_event(1))

    # Mediator→outcome slope linkage
    b_late_over = numpyro.sample("b_late_over", dist.Normal(0.,0.7))

    # Compose slopes
    beta_late = beta_late_base + jnp.dot(GA, thetaA_late) +
(jnp.dot(GB, thetaB_late) if nB>0 else 0.0) + jnp.dot(C, eta_ctrl_late)
    beta_over = beta_over_base + b_late_over*beta_late + jnp.dot(GA,
phiA_over) + (jnp.dot(GB, phiB_over) if nB>0 else 0.0) + jnp.dot(C,
delta_ctrl_over)

    numpyro.deterministic("beta_late", beta_late)
    numpyro.deterministic("beta_over", beta_over)

    # TVCs (naming aligned): k_* and rho_* ~ HalfNormal(1) for rates
    k_op = numpyro.sample("k_over_pulse",    dist.Normal(0.,0.5))
    k_or = numpyro.sample("k_over_recovery", dist.Normal(0.,0.5))
    k_lp = numpyro.sample("k_late_pulse",    dist.Normal(0.,0.5))
    k_lr = numpyro.sample("k_late_recovery", dist.Normal(0.,0.5))

    rho_over = numpyro.sample("rho_over_recovery",
dist.HalfNormal(1.0))
    rho_late = numpyro.sample("rho_late_recovery",
dist.HalfNormal(1.0))

    rec_over = 1.0 - jnp.exp(-rho_over * recovery)
    rec_late = 1.0 - jnp.exp(-rho_late * recovery)

    def obs_mu(a, b, tp, tr):
        mu = a[:,None] + b[:,None]*t[None,:]
        mu = mu + tp + tr
        return mu

    mu_over = obs_mu(alpha_over, beta_over, k_op*pulse, k_or*rec_over)
    mu_late = obs_mu(alpha_late, beta_late, k_lp*pulse,  k_lr*rec_late)

    numpyro.sample("Y_over", dist.Normal(mu_over, sigma_over),
obs=Y_over)
    numpyro.sample("Y_late", dist.Normal(mu_late, sigma_late),
obs=Y_late)

# ============================================================
# 6) Run helper with LOO/WAIC on Y_over
# ============================================================
def run_numpyro(model_fn, name: str, save_dir: str, var_name_for_ic:
str, **kwargs):
```

```
    os.makedirs(save_dir, exist_ok=True)
    rng_key = random.PRNGKey(SEED)
    kernel = NUTS(model_fn, target_accept_prob=TARGET_ACCEPT)
    mcmc = MCMC(kernel, num_warmup=WARMUP, num_samples=SAMPLES,
num_chains=CHAINS, progress_bar=True)
    mcmc.run(rng_key, **kwargs)

    # Divergence check
    extra = mcmc.get_extra_fields(group_by_chain=True)
    div = extra.get("diverging", None)
    if div is not None:
        rate = float(np.mean(np.asarray(div)))
        print(f"[{name}] Divergences: total={int(np.sum(div))},
rate={rate:.2%}")

    idata = az.from_numpyro(mcmc)

    # Add log_likelihood for IC
    samples_grouped = mcmc.get_samples(group_by_chain=True)
    ll_parts = log_likelihood(model_fn, samples_grouped, batch_ndims=2,
**kwargs)
    if var_name_for_ic not in ll_parts:
        raise KeyError(f"log_likelihood missing '{var_name_for_ic}', got
{list(ll_parts.keys())}")
    idata.log_likelihood = xr.Dataset({var_name_for_ic:
(("chain","draw","obs_dim_0","obs_dim_1"),

np.asarray(ll_parts[var_name_for_ic]))})

    # LOO & WAIC
    loo = az.loo(idata, var_name=var_name_for_ic, pointwise=False)
    waic = az.waic(idata, var_name=var_name_for_ic, pointwise=False)

    def _get(obj, nm, default=np.nan):
        return getattr(obj, nm, default)

    pd.DataFrame([{
        "ELPD LOO": _get(loo, "elpd_loo"),
        "S.E.":     _get(loo, "elpd_loo_se"),
        "LOOIC":    _get(loo, "looic"),
        "WAIC":     _get(waic, "waic"),
        "WAIC S.E.":_get(waic, "waic_se"),
    }]).to_csv(os.path.join(save_dir, f"fit_{name}.csv"), index=False,
encoding="utf-8-sig")

    # Save idata & summary
    az.to_netcdf(idata, os.path.join(save_dir, f"idata_{name}.nc"))
    try:
```

```
        summ = az.summary(idata, round_to=None, extend=True)
    except TypeError:
        summ = az.summary(idata, round_to=None)
    summ.reset_index().rename(columns={"index":"param"}).to_csv(
        os.path.join(save_dir, f"summary_{name}.csv"),
        index=False, encoding="utf-8-sig"
    )
    return idata

# ============================================================
# 7) Fit R (joint)
# ============================================================
DIR_R = OUT_ROOT
baseline_id_A = A_levels.index("CON") if "CON" in A_levels else 0
print(f"\n[R] groups A: {A_levels} | baseline A =
{A_levels[baseline_id_A]}")
print(f"[R] groups B: {B_levels}")
print(f"[R] baseline B (auto = most frequent) index = {baseline_id_B if
len(B_levels)>0 else 'NA'} | label = {B_levels[baseline_id_B] if
len(B_levels)>0 else 'NA'}")

idata_R = run_numpyro(
    model_R_full, name="R_full", save_dir=DIR_R,
var_name_for_ic="Y_over",
    Y_over=jnp.asarray(Y_over_z), Y_late=jnp.asarray(Y_late_z),
    C=jnp.asarray(C), GA=jnp.asarray(GA), GB=jnp.asarray(GB),
    t=t, pulse=jnp.asarray(pulse), recovery=jnp.asarray(recovery),
    baseline_id_A=int(baseline_id_A), baseline_id_B=int(baseline_id_B)
)

# ============================================================
# 8) Quick audit table (optional)
# ============================================================
def hdi_1d(samples: np.ndarray, prob: float=0.89):
    h = np.asarray(az.hdi(np.asarray(samples).reshape(-1),
hdi_prob=prob)).ravel()
    return float(h[0]), float(h[1])

def get_param(idata, name):
    if name in idata.posterior:
        return idata.posterior[name].values
    return None

rows = []

def add_block(vec, labels, prefix):
    if vec is None or len(labels)==0: return
    v = vec.reshape(-1, vec.shape[-1])
```

```
    for i, lab in enumerate(labels):
        m = float(v[:,i].mean()); sd=float(v[:,i].std(ddof=1))
        lo, hi = hdi_1d(v[:,i], 0.89)
        rows.append({"param": f"{prefix}[{lab}]", "mean": m, "sd": sd,
"hdi89_low": lo, "hdi89_high": hi})

add_block(get_param(idata_R,"phiA_over"),   A_levels, "phiA_over (A
group → OVERALL slope)")
add_block(get_param(idata_R,"phiB_over"),   B_levels, "phiB_over (B
direction → OVERALL slope)")
add_block(get_param(idata_R,"thetaA_late"), A_levels, "thetaA_late (A
group → LATE slope)")
add_block(get_param(idata_R,"thetaB_late"), B_levels, "thetaB_late (B
direction → LATE slope)")

b = get_param(idata_R,"b_late_over")
if b is not None:
    v = b.reshape(-1)
    rows.append({
        "param":"b_late_over (LATE slope → OVERALL slope)",
        "mean": float(v.mean()), "sd": float(v.std(ddof=1)),
        "hdi89_low": hdi_1d(v,0.89)[0], "hdi89_high": hdi_1d(v,0.89)[1]
    })

pd.DataFrame(rows).to_csv(os.path.join(DIR_R, "effects_R_full.csv"),
index=False, encoding="utf-8-sig")

# ===========================================================
# 9) Stats helpers for Table 3/4 (pd, ROPE, etc.) + write tables
# ===========================================================
def flat(x): return np.asarray(x).reshape(-1)

def hdi89_vals(x):
    lo,hi = az.hdi(flat(x), hdi_prob=0.89)
    return float(lo), float(hi)

def pdirection_vals(x):
    s = flat(x); return float(max((s>0).mean(), (s<0).mean()))

def rope_stats(x, rope=(-0.1, 0.1)):
    s = flat(x)
    b = float((s < rope[0]).mean())
    w = float(((s >= rope[0]) & (s <= rope[1])).mean())
    a = float((s > rope[1]).mean())
    return b, w, a

def write_table(save_csv, rows):
```

```
    pd.DataFrame(rows).to_csv(save_csv, index=False, encoding="utf-8-
sig")

def make_table3_overall_R(idata, A_levels, B_levels, save_csv: str,
ctrl_names=("Mean","Unemp_Rate","P_Density")):
    rows = []
    for nm, lbls in [("phiA_over", A_levels), ("phiB_over", B_levels)]:
        arr = get_param(idata, nm)
        if arr is not None and len(lbls)>0:
            for j, lab in enumerate(lbls):
                s = arr[..., j]; mean=float(np.mean(s));
lo,hi=hdi89_vals(s); pdv=pdirection_vals(s)
                b,w,a = rope_stats(s)
                rows.append({"param": f"{nm}[{lab}]", "mean":mean,
"hdi89_low":lo, "hdi89_high":hi, "pd":pdv, "rope_below":b,
"rope_within":w, "rope_above":a})
    # mediator
    b_med = get_param(idata,"b_late_over")
    if b_med is not None:
        s=b_med; mean=float(np.mean(s)); lo,hi=hdi89_vals(s);
pdv=pdirection_vals(s); b,w,a=rope_stats(s)
        rows.append({"param":"b_late_over", "mean":mean, "hdi89_low":lo,
"hdi89_high":hi, "pd":pdv, "rope_below":b, "rope_within":w,
"rope_above":a})
    # TVC & recovery (OVERALL)
    for p in
["k_over_pulse","k_over_recovery","rho_over_recovery","sigma_over"]:
        arr = get_param(idata, p)
        if arr is not None:
            s=arr; mean=float(np.mean(s)); lo,hi=hdi89_vals(s);
pdv=pdirection_vals(s); b,w,a=rope_stats(s)
            rows.append({"param":p, "mean":mean, "hdi89_low":lo,
"hdi89_high":hi, "pd":pdv, "rope_below":b, "rope_within":w,
"rope_above":a})
    # controls on OVERALL
    delta = get_param(idata,"delta_ctrl_over")
    if delta is not None:
        for k, nm in enumerate(ctrl_names):
            if k<delta.shape[-1]:
                s=delta[...,k]; mean=float(np.mean(s));
lo,hi=hdi89_vals(s); pdv=pdirection_vals(s); b,w,a=rope_stats(s)
                rows.append({"param":f"delta_ctrl_over[{nm}]",
"mean":mean, "hdi89_low":lo, "hdi89_high":hi, "pd":pdv, "rope_below":b,
"rope_within":w, "rope_above":a})
    write_table(save_csv, rows)

def make_table4_late_R(idata, A_levels, B_levels, save_csv: str,
ctrl_names=("Mean","Unemp_Rate","P_Density")):
```

```
    rows = []
    for nm, lbls in [("thetaA_late", A_levels), ("thetaB_late",
B_levels)]:
        arr = get_param(idata, nm)
        if arr is not None and len(lbls)>0:
            for j, lab in enumerate(lbls):
                s = arr[..., j]; mean=float(np.mean(s));
lo,hi=hdi89_vals(s); pdv=pdirection_vals(s)
                b,w,a = rope_stats(s)
                rows.append({"param": f"{nm}[{lab}]", "mean":mean,
"hdi89_low":lo, "hdi89_high":hi, "pd":pdv, "rope_below":b,
"rope_within":w, "rope_above":a})
    # TVC & recovery (LATE)
    for p in
["k_late_pulse","k_late_recovery","rho_late_recovery","sigma_late"]:
        arr = get_param(idata, p)
        if arr is not None:
            s=arr; mean=float(np.mean(s)); lo,hi=hdi89_vals(s);
pdv=pdirection_vals(s); b,w,a=rope_stats(s)
            rows.append({"param":p, "mean":mean, "hdi89_low":lo,
"hdi89_high":hi, "pd":pdv, "rope_below":b, "rope_within":w,
"rope_above":a})
    # controls on LATE
    eta = get_param(idata,"eta_ctrl_late")
    if eta is not None:
        for k, nm in enumerate(ctrl_names):
            if k<eta.shape[-1]:
                s=eta[...,k]; mean=float(np.mean(s));
lo,hi=hdi89_vals(s); pdv=pdirection_vals(s); b,w,a=rope_stats(s)
                rows.append({"param":f"eta_ctrl_late[{nm}]", "mean":mean,
"hdi89_low":lo, "hdi89_high":hi, "pd":pdv, "rope_below":b,
"rope_within":w, "rope_above":a})
    write_table(save_csv, rows)

make_table3_overall_R(idata_R, A_levels, B_levels, os.path.join(DIR_R,
"table3_R_full_aligned.csv"))
make_table4_late_R(  idata_R, A_levels, B_levels, os.path.join(DIR_R,
"table4_R_full_aligned.csv"))

# ===========================================================
# 10) Trajectory utilities (naming aligned)
# ===========================================================
def posterior_group_traj(idata, idx_arr: np.ndarray, t_vec: np.ndarray,
                      scaler_key="Overall") -> Tuple[np.ndarray,
np.ndarray]:
    """Return (median, hdi89[T,2]) in raw units for selected LA
indices."""
    if (idx_arr is None) or (len(idx_arr)==0):
```

```
        return None
    A0 = idata.posterior["alpha_over"].values  # (chain, draw, N)
    B0 = idata.posterior["beta_over"].values   # (chain, draw, N)
    CD = A0.shape[0]*A0.shape[1]
    A = A0.reshape(CD, A0.shape[2])[:, idx_arr].mean(axis=1)  # (CD,)
    B = B0.reshape(CD, B0.shape[2])[:, idx_arr].mean(axis=1)  # (CD,)
    y_draws = A[:,None] + B[:,None]*t_vec[None,:]           # z-units
    mu = scaler[scaler_key]["mean"]; sd = scaler[scaler_key]["std"]
    y_draws = y_draws*sd + mu
    med = np.median(y_draws, axis=0)
    hdi = np.asarray(az.hdi(y_draws, hdi_prob=0.89))         # (T,2)
    return med, hdi

def plot_trajectories_with_ref(idata, idx_dict: Dict[str,np.ndarray],
levels: List[str],
                               years: List[int], title: str, save_path: str,
                               baseline_label_for_color: str=None,
                               ref_label: str=None, ref_median=None,
ref_hdi=None):
    """Plot medians and 89% HDIs by group; optionally overlay a
reference line."""
    plt.figure(figsize=(9,5.2))
    # color assignment
    colors = {}
    if baseline_label_for_color and (baseline_label_for_color in
levels):
        colors[baseline_label_for_color] = CON_DEEP_GRAY
    others = [lab for lab in levels if lab != baseline_label_for_color]
    for i, lab in enumerate(others):
        colors[lab] = PALETTE[i % len(PALETTE)]

    # plot groups
    for lab in levels:
        idx_arr = idx_dict.get(lab, np.array([], dtype=int))
        if idx_arr.size == 0:
            continue
        med, h = posterior_group_traj(idata, idx_arr, t_c, "Overall")
        c = colors.get(lab, "#444444")
        plt.plot(years, med, lw=2.2, color=c, label=lab)
        plt.fill_between(years, h[:,0], h[:,1], alpha=0.18, color=c)

    # reference overlay
    if (ref_label is not None) and (ref_median is not None) and
(ref_hdi is not None):
        plt.plot(years, ref_median, lw=2.8, color=CON_DEEP_GRAY,
label=ref_label)
        plt.fill_between(years, ref_hdi[:,0], ref_hdi[:,1],
color=CON_DEEP_GRAY, alpha=0.12)
```

```
    plt.ylim(*YLIM); plt.xlabel("Year"); plt.ylabel("CQC Overall
(raw)")
    plt.title(title)
    plt.legend(frameon=False, ncol=2)
    plt.tight_layout()
    plt.savefig(save_path, dpi=220, bbox_inches="tight"); plt.close()

# ==========================================================
# 11) Plot A/B trajectories
# ==========================================================
la_A_cat = pd.Categorical([A_label_map.get(la, None) for la in
ons_list], categories=A_levels)
idx_A = {g: np.where(la_A_cat==g)[0] for g in A_levels}

la_B_cat = pd.Categorical([B_label_map.get(la, None) for la in
ons_list], categories=B_levels)
idx_B = {g: np.where(la_B_cat==g)[0] for g in B_levels}

# A: long-run (baseline color = CON deep gray)
plot_trajectories_with_ref(
    idata_R, idx_A, A_levels, YEARS,
    title="Robustness R — OVERALL by long-run party (A)",
    save_path=os.path.join(DIR_R, "traj_R_A_OVERALL_aligned.png"),
    baseline_label_for_color="CON",
    ref_label=None, ref_median=None, ref_hdi=None
)

# A:CON reference for B plot
idx_A_CON = idx_A.get("CON", np.array([], dtype=int))
A_CON_ref = (posterior_group_traj(idata_R, idx_A_CON, t_c, "Overall")
if idx_A_CON.size>0 else None)
ref_med, ref_hdi = (A_CON_ref if A_CON_ref is not None else (None,
None))

# B: transitions + A:CON reference overlay
plot_trajectories_with_ref(
    idata_R, idx_B, B_levels, YEARS,
    title="Robustness R — OVERALL by transition direction (B) + A:CON
ref",
    save_path=os.path.join(DIR_R, "traj_R_B_OVERALL_aligned.png"),
    baseline_label_for_color=None,
    ref_label=("A: CON (reference)" if A_CON_ref is not None else
None),
    ref_median=ref_med, ref_hdi=ref_hdi
)

# ==========================================================
```

```
# 12) Figure 1: Forest plots (OVERALL / LATE) — names aligned
# ==========================================================
def _forest_plot(df_plot: pd.DataFrame, title: str, save_path: str,
unit_label: str="z"):
    """Plot mean and 89% HDI for selected parameters."""
    if df_plot is None or df_plot.empty:
        print(f"[forest] {title} — No drawing parameters, skipping.");
return
    dfp = df_plot.copy()
    dfp["y"] = np.arange(len(dfp))[::-1]
    plt.figure(figsize=(10.5, max(2.5, 0.5*len(dfp)+1.2)))
    plt.hlines(dfp["y"], dfp["hdi_low"], dfp["hdi_high"], lw=3,
color="#444444")
    plt.plot(dfp["mean"], dfp["y"], "o", ms=5, color="#111111")
    plt.axvline(0, lw=1.2, ls="--", color="#666666")
    plt.yticks(dfp["y"], dfp["label"])
    plt.xlabel(f"Effect ({unit_label})")
    plt.title(title)
    plt.tight_layout()
    plt.savefig(save_path, dpi=220, bbox_inches="tight")
    plt.close()

def _arr_to_rows(arr, labels, omit_idx=None, prefix=None):
    rows = []
    if arr is None: return rows
    L = len(labels)
    for j in range(L):
        if (omit_idx is not None) and (j == omit_idx):
            continue
        s = arr[..., j]
        mean = float(np.mean(s)); lo,hi = hdi89_vals(s)
        lab = f"{prefix}: {labels[j]}" if prefix else labels[j]
        rows.append({"label": lab, "mean": mean, "hdi_low": lo,
"hdi_high": hi})
    return rows

def build_forest_overall_R(idata, A_levels, B_levels, baseline_id_A,
baseline_id_B, ctrl_names=("Mean","Unemp_Rate","P_Density")):
    rows = []
    rows += _arr_to_rows(get_param(idata,"phiA_over"), A_levels,
omit_idx=baseline_id_A, prefix="A→OVERALL")
    if len(B_levels)>0:
        rows += _arr_to_rows(get_param(idata,"phiB_over"), B_levels,
omit_idx=baseline_id_B, prefix="B→OVERALL")
    b = get_param(idata,"b_late_over")
    if b is not None:
        mean, (lo,hi) = float(np.mean(b)), hdi89_vals(b)
```

```
        rows.append({"label":"Mediator b(LATE→OVERALL)", "mean":mean,
"hdi_low":lo, "hdi_high":hi})
    delta = get_param(idata,"delta_ctrl_over")
    if delta is not None:
        for k, nm in enumerate(ctrl_names):
            s = delta[...,k]; mean=float(np.mean(s)); lo,hi=hdi89_vals(s)
            rows.append({"label":f"Control[{nm}] → OVERALL slope",
"mean":mean, "hdi_low":lo, "hdi_high":hi})
    for nm, lab in [("k_over_pulse","Pulse(2020) amplitude"),
                    ("k_over_recovery","Recovery amplitude k"),
                    ("rho_over_recovery","Recovery rate ρ")]:
        s = get_param(idata, nm)
        if s is not None:
            mean=float(np.mean(s)); lo,hi=hdi89_vals(s)
            rows.append({"label":lab, "mean":mean, "hdi_low":lo,
"hdi_high":hi})
    return pd.DataFrame(rows)

def build_forest_late_R(idata, A_levels, B_levels, baseline_id_A,
baseline_id_B, ctrl_names=("Mean","Unemp_Rate","P_Density")):
    rows = []
    rows += _arr_to_rows(get_param(idata,"thetaA_late"), A_levels,
omit_idx=baseline_id_A, prefix="A→LATE")
    if len(B_levels)>0:
        rows += _arr_to_rows(get_param(idata,"thetaB_late"), B_levels,
omit_idx=baseline_id_B, prefix="B→LATE")
    eta = get_param(idata,"eta_ctrl_late")
    if eta is not None:
        for k, nm in enumerate(ctrl_names):
            s = eta[...,k]; mean=float(np.mean(s)); lo,hi=hdi89_vals(s)
            rows.append({"label":f"Control[{nm}] → LATE slope",
"mean":mean, "hdi_low":lo, "hdi_high":hi})
    for nm, lab in [("k_late_pulse","Pulse(2020) amplitude (LATE)"),
                    ("k_late_recovery","Recovery amplitude k (LATE)"),
                    ("rho_late_recovery","Recovery rate ρ (LATE)")]:
        s = get_param(idata, nm)
        if s is not None:
            mean=float(np.mean(s)); lo,hi=hdi89_vals(s)
            rows.append({"label":lab, "mean":mean, "hdi_low":lo,
"hdi_high":hi})
    return pd.DataFrame(rows)

forest_over_R = build_forest_overall_R(idata_R, A_levels, B_levels,
baseline_id_A, baseline_id_B)
_forest_plot(forest_over_R, "Key effects on OVERALL (z; 89% HDI) —
Robustness R (aligned)",
             os.path.join(DIR_R, "fig1_overall_R_full_aligned.png"),
unit_label="z")
```

```
forest_late_R = build_forest_late_R(idata_R, A_levels, B_levels,
baseline_id_A, baseline_id_B)
_forest_plot(forest_late_R, "Key effects on LATE (z; 89% HDI) —
Robustness R (aligned)",
             os.path.join(DIR_R, "fig1_late_R_full_aligned.png"),
unit_label="z")

print("\n[Done] Robustness (R, aligned to Code1) finished. Files saved
in:", OUT_ROOT)
print("Generated files:")
print("  - traj_R_A_OVERALL_aligned.png, traj_R_B_OVERALL_aligned.png
(Figure 2)")
print("  - fig1_overall_R_full_aligned.png,
fig1_late_R_full_aligned.png  (Figure 1)")
print("  - effects_R_full.csv, table3_R_full_aligned.csv,
table4_R_full_aligned.csv")
gc.collect()
```

University of Bristol Research Ethics Application

Investigator information

Application Submitter Details

Title

Mr

First Name

Sigao

Surname

Li

Faculty

Arts, Law and Social Sciences

Department

School of Management - Business School

School

Telephone

Email

ez24579@bristol.ac.uk

Preferred Name or Also Known As

Faculty

Social Sciences and Law

School / Department / Centre

University of Bristol Business School

Are you a student submitting this ethics application as part of your degree qualification?

Yes

Please declare your level of study

Taught Masters

Supervisor Contact Details

Title

Dr

First Name

Wen

Surname

Zhang

Department

Management

Faculty

Faculty of Social Sciences and Law

Email

wen.zhang@bristol.ac.uk

Title

Dr

First Name

Sunil

Surname

Tiwari

Department

Management

Faculty

Faculty of Social Sciences and Law

Email

sunil.tiwari@bristol.ac.uk

Are you an academic member of staff submitting an ethics application on behalf of a student(s) as part of their degree qualification?

No

Second Supervisor Details. If University of Bristol, please provide their full name and title.

If external to the University of Bristol, please provide their name, organisation details, email address and telephone number.

Dr. Minhao Zhang, Associate Professor in Operations and Supply Chain Management, University of Bristol Business School, minhao.zhang@bristol.ac.uk

Please provide details of any other researchers/collaborators involved in the study.

Dr. Mike Tse, Professor in Operations and Supply Chain Management, Cardiff Business School, TseM1@cardiff.ac.uk, +44 29225 11764

Are you submitting this ethics application on behalf of another researcher?

No

Ethics Committee Review

Has or will your research be submitted to another research ethics committee for research involving human participants, their tissue and or data?

- ☐ Yes
- ☑ No

Important Information - Please note:

It is extremely important that you select the **correct Research Ethics Committee (REC)** to review your research ethics application.

The REC selected, will determine the questions you are asked to complete on this online form and the research ethics committee that will review your research ethics application.

Please note, if you select the incorrect ethics committee, this may delay the review of your ethics application as your ethics application will need to be returned to you so that you can select the correct REC and complete the relevant questions on the online form.

If you are unsure of the correct research ethics committee to select please contact research-ethics@bristol.ac.uk

Please select the Research Ethics Committee (REC) to review your research ethics application:

Business School Research Ethics Committee

To proceed to the next page select 'Next' in the Actions tiles.

To save your application for completion and submission at a later date please select 'Save' in the Actions tiles.

Lead questions

Does your research involve any of the following? Tick all that apply

- ☐ Patients, clients or carers of an NHS Trust or Social Care organisation
- ☐ Solely staff or premises of an NHS Trust or Social Care organisation?
- ☐ Staff, residents or premises of one or more care homes
- ☐ Participants who are lacking or have diminished mental capacity
- ☐ Staff, inmates or premises of one or more prisons
- ☐ Staff, participants or premises of one or more local authority departments
- ☑ None of the above

To proceed to the next page select 'Next' in the Actions tiles.

To save your application for completion and submission at a later date please select 'Save' in the Actions tiles.

Brief study outline

Study Title

The Impact of Changes in Party Political Control in England on Care Home Quality

Brief Project Outline (up to approximately 300 words in plain English)

How do changes in party political control of local authorities in England affect care home quality—measured jointly by (i) Care Quality Commission (CQC) inspection ratings, (ii) Google Maps review sentiment, and (iii) government public spending on adult care—between 2015 and 2024, and how big is the impact of changes in government?

Estimated dates of research data collection

Please note that you must not start data collection for your research project until you have received a formal favourable ethics opinion from the appropriate Research Ethics Committee (REC). Please factor in the research ethics review timescales when selecting the estimated dates for research data collection.

Anticipated Start Date

23/06/2025

Anticipated End Date

28/08/2025

To proceed to the next page select 'Next' in the Actions tiles.

To save your application for completion and submission at a later date please select 'Save' in the Actions tiles.

Data collection method

BUSINESS SCHOOL RESEARCH ETHICS
APPLICATION FORM: TAUGHT POSTGRADUATE STUDENTS

This form must be completed for each piece of research carried out by all taught post-graduate students in the Business School.

Students should discuss their proposed research with their supervisors who will then approve and sign this form before forwarding to the relevant dissertation convenor (or in some cases unit convenor or programme director) who will approve the form on behalf of the Business School REC when they are happy with the contents.

Failure to get approval prior to conducting any fieldwork may result in the University taking action for research misconduct – the outcome of such action may be that you are unable to submit your fieldwork findings for assessment and **your degree may not be awarded.**

Once your study is approved, you must follow the plan described in this form. You should remember that ethics is an on-going process, ie your ethical thinking is not 'done' when your form is signed. It is about how you act as a researcher. You should remain reflexive throughout the research process and think about how the research is impacting on your participants and yourself. You should refer to this completed form throughout your research process to make sure you are remaining within your ethical approval. If you wish to change your research plan, then you must discuss this with your supervisor. If the change is very small your supervisor can approve the change. However, if the change is more significant, you will need to ask for an amendment to your ethical approval. Your supervisor and dissertation convenor must approve this change in writing. If you do not get approval for changes, then you won't have ethical approval for the change, and it may result in the University taking action for research misconduct.

Terminology used in this form:

1. **Primary research** includes any research that collects new data such as interviews, focus groups, observations, online surveys, new data collected via a social media post etc.
2. **Secondary analysis/literature review** relates to the re-analysis of data that already exists such as analysis of publicly available documents or tv programmes, analysis of existing social media posts, reviews systematic or otherwise, or statistical analysis of analysis of publicly available datasets etc.

Please select the method of data collection relevant to your research. Tick all that apply

- ☐ Primary research data collection
- ☑ Secondary data analysis

Please provide your student number

2638795

To proceed to the next page select 'Next' in the Actions tiles.

To save your application for completion and submission at a later date please select 'Save' in the Actions tiles.

Study design and background

Research involving humans by all academic and related Staff and Students in the University of Bristol Business School is subject to the standards set out in the University of Bristol Ethics of Research Policy and Procedure which can be found at: http://www.bristol.ac.uk/red/research-governance/practice-training/researchethicspolicy.pdf

It is a requirement prior to the commencement of all funded and non-funded research that this form be completed and submitted to the School's Research Ethics Committee (REC). The REC will be responsible for issuing certification that the research meets acceptable ethical standards and will, if necessary, require changes to the research methodology or reporting strategy. It is a requirement that prior to the commencement of all funded and non-funded research that this form be completed and submitted to the School's Research Ethics Committee (REC). The REC will be responsible for issuing certification that the research meets acceptable ethical standards and will, if necessary, require changes to the research methodology or reporting strategy.

For those intending to carry out secondary analysis of data or a systematic review:

Please provide details of where you are getting your data set from and how you will use this data. Data sets must be stored on the University of Bristol server.

What sources / secondary datasets will you use?

I intend to analyze social and economic data from the UK Data Service, including:

Care Quality Commission (CQC) inspection report
Google Maps review of care homes
Government spending on adult care
NHS spending on adult care
England election results

Where will you get these data from (e.g. ESRC Data Archive, systematic literature review, document archive). Please describe your selection criteria and how you will locate/access the data?

CQC data:
https://www.cqc.org.uk/about-us/transparency/using-cqc-data

Google Maps Reviews:
https://console.apify.com/actors/Xb8osYTtOjlsgI6k9/input?addFromActorId=Xb8osYTtOjlsgI6k9

Local authority revenue expenditure and financing:
https://www.gov.uk/government/collections/local-authority-revenue-expenditure-and-financing#2016-to-2017

Adult Social Care Activity and Finance Report:
https://digital.nhs.uk/data-and-information/publications/statistical/adult-social-care-activity-and-finance-report

Elections data:
https://commonslibrary.parliament.uk/tag/elections-data/

If necessary, how will you obtain permission to use these data? This would apply to data sets where it is usual for the researcher to sign an end user licence.

The dataset I intend to use is open access. Users can access open data collections without registration. These collections are subject to Open Data Licences and conditions for example the Open Government Licence or Creative Commons Licence.

How will you analyse the data?

I will use appropriate statistical methods, such as descriptive statistics, causal analysis, or hypothesis testing, to address my research questions.

What ethical issues will you consider when undertaking secondary data analysis? i.e. will you consider the quality of the papers/programmes etc reviewed?

I will respect the integrity of the original data, maintain its confidentiality, and avoid causing harm to individuals or groups. I will ensure that the data is used in accordance with the original consent, manage the data securely, and appropriately attribute its source.

Clearly state any potential risk to you as the researcher and how you will address this risk.

N/A

Supporting Information

Please provide any additional information relevant to your secondary data research.

N/A

Please upload any supporting information relevant to your secondary data research.

Supporting Information

To be completed by your named supervisor

This free text box below is to be completed by your supervisor.

Your form must now be reviewed by your supervisor confirming that any ethics related issues relating to your proposed research have been addressed and your ethics application is complete.

So that your supervisor can confirm this, please follow these steps:

1. If you have not already done so, share your ethics application with your supervisor by selecting the 'Share' button on the left hand side of the form.
2. Input your supervisor email address in the pop-up box, and grant them 'Read' and 'Write' and 'submit' access.
3. You will then need to email your supervisor separately with the following email requesting them to complete this section:

    "Dear [Supervisor Name]

    My research ethics application is now ready for you to review and confirm that the ethics related issues relating to my proposed research have been addressed and my ethics application is complete. Can you please click on the link below, complete the free text in the ethics application form to confirm this.

    [Insert application URL]

    All the best"

Once this section has been completed by your supervisor you please select on the next page, and begin the Ethics Academic Checker declaration process on the next page.

approved

Supervisor Guidance Note:

Please state your name and the date of signature.

wen zhang, 21/June/2025

Business School Ethics Academic Checker Declaration

Guidance Note

In order to ensure that your ethics application is submitted successfully please follow the steps below.

Your form must now be reviewed by your assigned Academic Ethics Checker. They will determine whether your ethics application requires further review. So that your Academic Ethics Checker can complete their declaration the following steps must be completed.

1. Share your ethics application with the Academic Ethics Checker by selecting the 'Share' button on the left hand side of the form.
2. Input your Academic Ethics Checker's email address in the pop-up box, and grant them 'Read' and 'Write' access.
3. You will then need to email your Academic Ethics Checker separately with the following email requesting them to complete this section:

 "Dear [Ethics Academic Checker Name]

 My research ethics application is now ready for you to review and sign prior to submission. Can you please click on the link below, complete the Academic Ethics Checker declaration page and select the signature button.

 [Insert application URL]

 All the best"

Once the form has been signed by your named Academic Ethics Checker you will then need to complete the signature declaration on the next page.

Business School Ethics Checker Determination - Please tick the appropriate box below:

- [x] I have reviewed the student’s proposed research project, and believe that no further ethical review is necessary.
- [ ] I have reviewed the student’s proposed research project, and recommend that the application is reviewed by the Business School Research Ethics Committee

Academic Ethics Checker Signature

Signed: This form was signed by Dr Sunil Tiwari (sunil.tiwari@bristol.ac.uk) on 24/06/2025 00:01

Signatures

Applicant Signature

I confirm that my responses are complete and accurate.

I understand that any errors or omissions may cause a delay in the processing of my application.

Signed: This form was signed by Mr Sigao Li (ez24579@bristol.ac.uk) on 27/06/2025 01:50

Supervisor signature request

As the named supervisor please confirm:

1. I have read and reviewed this ethics application.
2. I am satisfied with the content and completeness of this ethics application.

Signed: This form was signed by Dr Wen Zhang (wen.zhang@bristol.ac.uk) on 27/06/2025 10:56

Submission Reminder

Please note - Once all signatures have been obtained. You must ensure you select the **Submit** button on the left hand side of the form.

If you are unsure whether your application has been submitted, please contact business-school-ethics@bristol.ac.uk.