CI Assignment – Z0195033 – Word Count: 2,983

Estimating the Causal Impact of Alcohol Minimum Unit Pricing on Hospitalisations in Scotland: A Difference in Differences Study.

Abstract

Scotland experiences disproportionately higher rates of alcohol-attributable health and social harms than other areas of the UK. To address this, the Scottish Government introduced minimum unit pricing (MUP) for alcohol on 1st May 2018, requiring every alcoholic drink sold to be priced at a minimum of 50p per unit. This report estimates whether MUP has plausibly caused a reduction in hospitalisations attributable to alcohol in Scotland. Local Authority level alcohol-attributable hospitalisations and socioeconomic deprivation data for both Scotland and England were used, with Scottish Local Authorities (treatment group) matched to English Local Authorities (control group) based on socioeconomic characteristics determined to be associated with levels of alcohol consumption and alcohol-attributable health harms from the literature. A Difference in Differences analysis was used to estimate the impact of MUP. MUP in Scotland was estimated to have plausibly reduced mean Local Authority alcohol-attributable hospitalisations per 100,000 residents per year by 44.5 (95% CI -78.0 to -11.0; p-value = 0.009), representing an estimated 6.2% (95% CI -10.9% to -1.5%; p-value = 0.009) or approximately 2,450 (95% CI -4,300 to -600; p-value = 0.009) reduction in total alcohol-attributable hospitalisations per year across Scotland.

1. Introduction

1.1 Background

Scotland experiences disproportionately higher rates of alcohol-attributable health and social issues than other areas of the UK (NRS, 2024). As part of the Scottish Governments strategy to reduce levels of alcohol consumption and the associated health and social harms, Scotland introduced minimum unit pricing (MUP) for alcohol on 1st May 2018, requiring every alcoholic drink sold to the public to be priced at a minimum of 50p per unit (PHS, 2018). This legislation set a legal minimum unit price of 50p below which alcohol is not permitted to be sold, differing from an increase in alcohol duty which can be circumvented by retailers' alcohol pricing strategy (Wilson et al, 2021). The MUP theory of change published by the Scottish Government (PHS, 2018) details the anticipated benefits of the legislation, primarily a reduction in alcohol-attributable health and social harms.

1.2 Theory

Scotland was the first country to implement MUP and therefore there is not a broad evidence base in the literature relating to its impact. However, MUP and increases in MUP have been shown to be associated with reduced alcohol consumption (Stockwell et al, 2012) and reduced alcohol-attributable health harms (Stockwell et al, 2013), in British Columbia, Canada. For Scotland, it has been estimated that 3 years post MUP introduction, total alcohol sales had reduced by 3%, plausibly due to MUP, driven by a reduction in off-trade alcohol sales i.e. alcohol sold in retailers for consumption off the premises (Giles et al, 2022). The introduction of MUP has previously been estimated to plausibly cause a 4.1% reduction in alcohol-attributable hospitalisations across Scotland (Wyper et al, 2023). This report aims to build on the current literature, assessing whether MUP has plausibly caused a reduction in alcohol-attributable hospitalisations, in line with the Scottish Governments MUP theory of change (PHS, 2018), and to build on the evidence from Wyper et al, 2023, using different data, methods, and formal causal inference as per the potential outcomes framework. Wyper et al, 2023 was the only paper found in a review of the literature to estimate the causal effect of MUP on health or social harms, specifically alcohol-attributable deaths and hospitalisations. Wyper et al used a controlled interrupted time series study to estimate the impact of MUP on alcohol-attributable hospitalisations, using individual level data, Scotland as the treatment group, and England as the control group. Individuals' postcodes were mapped to deprivation deciles between Scotland and England as a control, using Scottish Index of Multiple Deprivation (SIMD) (Scottish Government, 2016) and Index of Multiple Deprivation (IMD) (UK Government, 2010) data.

1.3 Methodological Overview

For this report, individual level alcohol-attributable hospitalisations data was not available, requiring applications to Public Health Scotland and NHS England for research access. Instead, Local Authority level alcohol-attributable hospitalisations data was used (PHS, 2024) (NHS, 2024). Instead of the whole of Scotland and England being used as treatment and control groups respectively, Mahalanobis distance was used to match Scottish Local Authorities to English Local Authorities based on socioeconomic characteristics (ONS, 2024), to create a control group. This was assessed to be a more robust method of incorporating socioeconomic characteristics than SIMD vs IMD deciles, due to SIMD and IMD containing different variables, different definitions and measurement methodologies, and different weighting to calculate their respective within country indices which themselves are not directly comparable given the varying levels of socioeconomic deprivation present in Scotland vs England (ONS, 2024). Additionally, this report will use pretreatment socioeconomic deprivation characteristics for matching from the same period to ensure matching characteristics are comparable and have not changed due to any phenomena effecting either or both groups over time, as opposed to matching using 2016 characteristics for Scotland versus 2010 characteristics for England as in Wyper et al, 2023.

This report uses the potential outcomes framework to define the possible states of Scotland, both with and without the introduction of MUP (Imbens and Rubin, 2011). The quantity this study aims to estimate is the casual impact of MUP on the mean Local Authority alcohol-attributable hospitalisations per 100,000 residents per year in Scotland. This outcome is the average treatment effect on the treated (ATT), with the treatment being MUP and the treated being Scottish Local Authorities. Y_1 and Y_0 indicate the potential outcomes if all Scottish Local Authorities were to implement MUP (Y_1) or not (Y_0) respectively. T represents the treatment, with T = 1 and T = 0 representing MUP having been implemented and not respectively.

$$ATT = \mathrm{E}[Y_1 - Y_0 \mid T = 1]$$

The ATT is therefore the expected value of the difference between the two potential outcomes Y_1 and Y_0, for treated units i.e. all Scottish Local Authorities. A Difference in Differences analysis was used to estimate the ATT, due to its applicability when data availability is sparse in terms of temporal range and resolution.

1.4 Results Overview

MUP in Scotland was estimated to have plausibly reduced mean Local Authority alcohol-attributable hospitalisations per 100,000 residents per year by 44.5 (95% CI -78.0 to -11.0; p-value = 0.009), representing an estimated 6.2% (95% CI -10.9% to -1.5%; p-value = 0.009) or approximately 2,450 (95% CI -4,300 to -600; p-value = 0.009) reduction in total alcohol-attributable hospitalisations per year across Scotland.

2. Data

2.1 Alcohol-Attributable Hospitalisations Data

Alcohol-attributable hospitalisations per year by Local Authority data was obtained for Scotland and England, from Public Health Scotland (PHS, 2024) and NHS England (NHS, 2024) respectively. This data included financial year (interval), Local Authority (nominal), and number of alcohol-attributable hospital stays per 100,000 residents (ratio, Scotland range: 273-1896, England range: 250-1150). There were no missing data values. The Scottish data covered financial years 1997/98 to 2022/23, with the English data available without a research application covering 2012/13 to 2019/20. This meant the study period was necessitated to cover at most 2012/13 to 2019/20, with only the last week of this period impacted by COVID-19 lockdowns. More granular data in terms of units measured, and in terms of temporal resolution was not available without research applications.

2.2 Local Authority Socioeconomic Characteristics Data

UK Annual Population Survey 2017 pretreatment data by Local Authority, age, and sex; and UK Annual Hours and Earnings Survey 2017 pretreatment data by Local Authority were obtained from the Office of National Statistics (ONS, 2024). The former contained Local Authority (nominal), economic activity rate for 16- to 64-year-old residents (ratio, range: 62.5-92.4), employment rate for 16- to 64-year-old residents (ratio, range: 57.4-90.9), unemployment rate for 16- to 64-year-old residents (ratio, range: 1.3-9.6), percentage of male residents (ratio, range: 47.4-53.7), percentage of 0- to 15-year-old residents (ratio, range: 7.7-27.1), percentage of 16- to 64-year-old residents (ratio, range: 53.2-77.1), and percentage of 65+ year old residents (ratio, range: 5.6-32.6); the latter contained median annual gross income (ratio, range: 21,089-43,105). Combined there were 190 missing data values, representing 6.7% of Local Authority characteristic values.

3. Methodology

3.1 Missing Data Imputation

Missing Local Authority characteristic data were imputed using multiple imputation by chained equations (MICE) (Van Buuren and Groothuis-Oudshoom, 2011) (CRAN, 2024), due to its ability to effectively incorporate the additional uncertainty from missing data (Azur, 2011). Predictive mean matching (PMM) was chosen for the individual imputation models due to the continuous and non-normally distributed nature of variables, for which the assumption of normality is required by linear regression (Zio and Guarnera, 2009). MICE using PMM was repeated five times to create five imputed dataset versions. All following analysis was repeated on each dataset version with results aggregated. Figure 3.1 shows imputed versus unimputed values for employment rate, all other imputed variables are shown in Annex Section A.

Figure 3.1 showing imputed values (red) compared to unimputed values (blue) for employment rate for 16- to 64-year-old residents, points jittered.

3.2 Local Authority Matching

English Local Authorities were matched to Scottish Local Authorities using socioeconomic characteristic data, detailed in Section 2.2. These socioeconomic variables were chosen due to having been found to be associated with levels of alcohol consumption and alcohol-attributable health harms across multiple meta-analyses (Brennar et al, 2015) (Jones et al, 2015) (Khamis et al, 2022) (Probst et al, 2014) (Temple et al, 1991). Mahalanobis distance (Mahalanobis, 1930) was chosen for matching as it is effective at accounting for correlations within matching data (see Figure 3.2.1), at matching variables with different scales, and at providing balanced matching in terms of treatment and control group covariates for studies with small sample sizes (Austin, 2011) (De Maesschalck et al, 2000) (Zhao, 2004). Matching resulted in 32 Scottish Local Authorities (treatment group) matched to the 32 English Local Authorities (control group), repeated for each imputed dataset version. Annex Section B details the specific matched Local Authorities for each imputed dataset run.

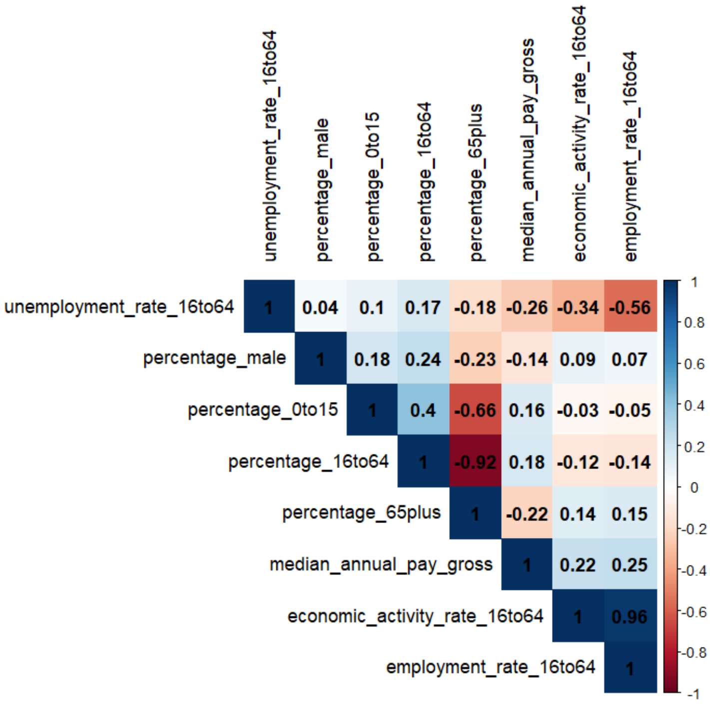

Figure 3.2.1 showing the Spearman rank correlation between each pair of Local Authority characteristics, both Scottish and English data.

The quality of matching was assessed to be reasonable in terms of the distributions of matching characteristics between the treatment and control group. Figure 3.2.2 visualises the distribution for a single variable from the first imputed dataset, with figures corresponding to all variables and other imputed datasets provided in Annex Section C.

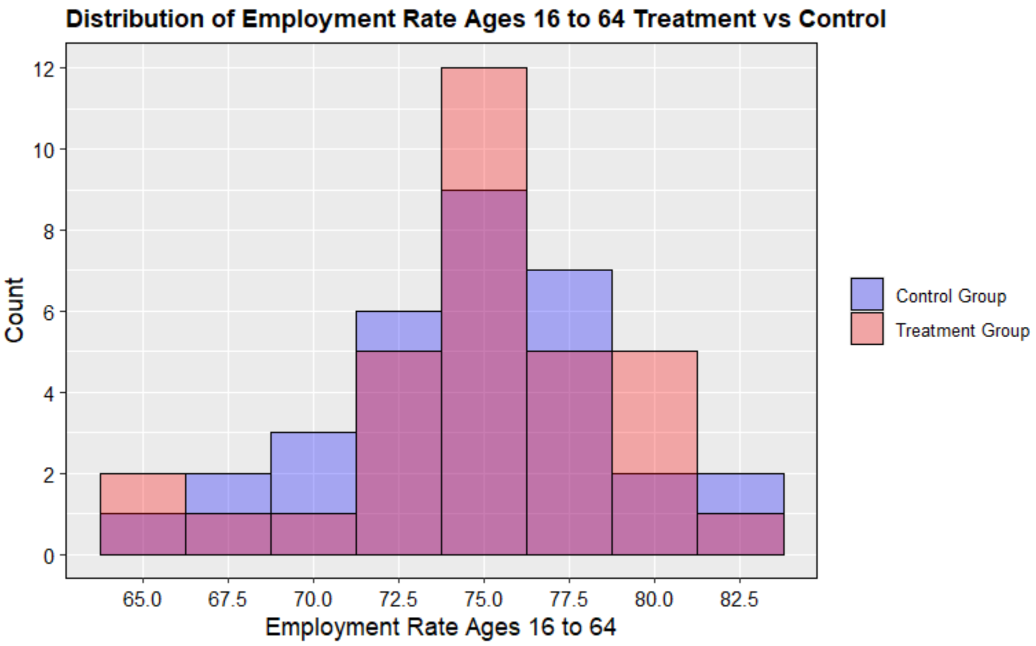

Figure 3.2.2 showing the distribution of employment rate for ages 16 to 64 between the control and treatment group.

3.3 Modelling

A Difference in Differences methodology was chosen due to the relatively small number of data points available as a result of both the limited English data availability, and the Scottish and English data being yearly by Local Authority. A two-year period from April 2018 to March 2020 was chosen for the analysis, using the closest pretreatment data to intervention and all available posttreatment data. The assumption of parallel pretreatment trends in terms of the treatment and control group outcome variable was assessed to be reasonable. Figure 3.3.1 shows the pretreatment trends for the first imputed dataset and Figure 3.3.2 shows for all imputed datasets, with full-size figures corresponding to the other four imputed datasets provided in Annex Section D, all figures show similar pretreatment trends between groups.

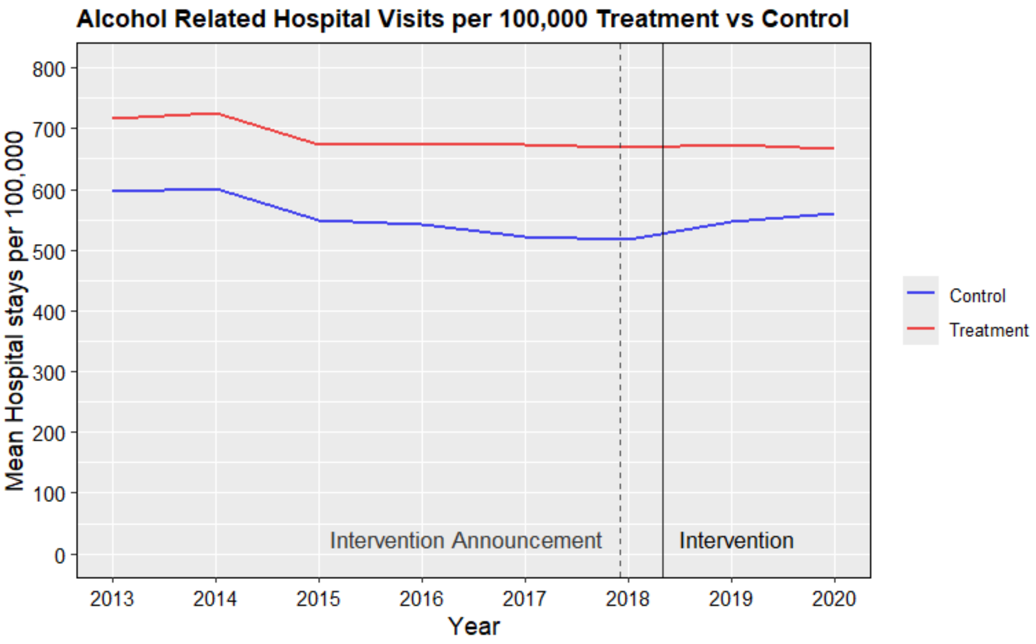

Figure 3.3.1 showing the mean Local Authority alcohol-attributable hospitalisations per 100,000 residents per year for the treatment and control group for imputed dataset 1.

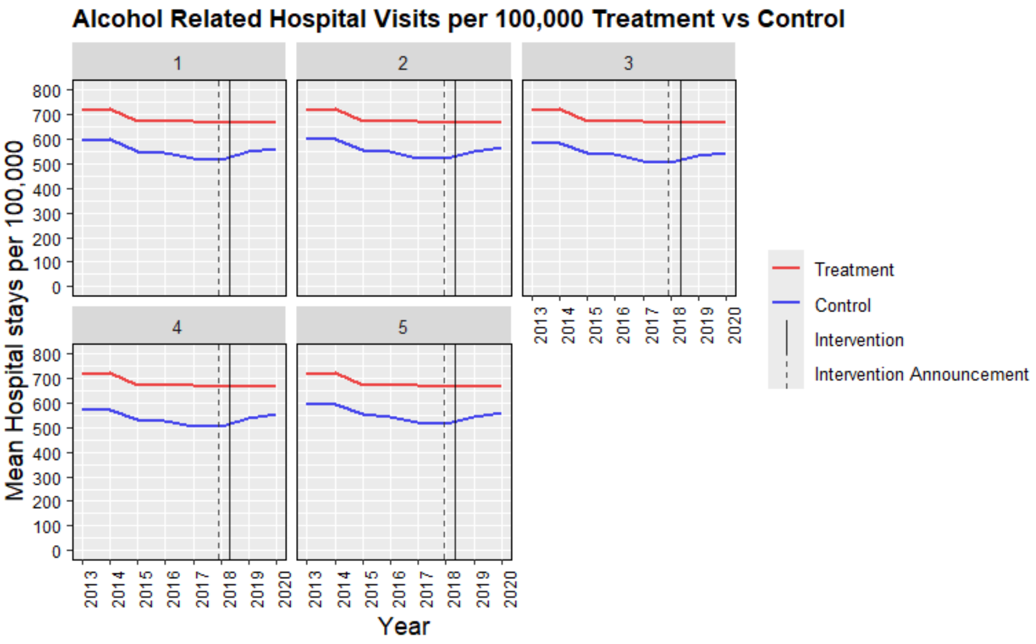

Figure 3.3.2 showing the mean Local Authority alcohol-attributable hospitalisations per 100,000 residents per year for treatment and control groups for all imputed datasets.

Bootstrapping was used to calculate the within run standard errors, 95% confidence intervals, and p-values due to the relatively small sample size, for which it has been shown to be effective for estimating said metrics (Efron and Tibshirani, 1986).

The entire matching, Difference in Differences, and bootstrapping process was repeated for each of the five imputed dataset versions. The ATT point estimates were averaged across the five runs to calculate the overall ATT estimate. Rubin's Rules were used to estimate the overall standard errors, 95% confidence intervals, and p-values, reflecting the additional uncertainty due to the imputed missing data (Rubin, 1987).

4. Findings

4.1 Main ATT Estimate

MUP in Scotland was estimated to have plausibly reduced mean Local Authority alcohol-attributable hospitalisations per 100,000 residents per year by 44.5 (95% CI -78.0 to -11.0; p-value = 0.009), representing an estimated 6.2% (95% CI -10.9% to -1.5%; p-value = 0.009) or approximately 2,450 (95% CI -4,300 to -600; p-value = 0.009) reduction in total alcohol-attributable hospitalisations per year across Scotland. The Difference in Differences methodology used suggests it is plausible that these effects can be causally attributed to MUP.

Estimated ATT, (95% CI)	**Estimated Percentage Change in Number of Hospitalisations per Year, (95% CI)**	**Estimated Change in Number of Hospitalisations per Year, (95% CI)**	**p-value**
-44.52 (-78.04 to -11.00)	-6.21% (-10.89% to -1.53%)	-2432 (-4264 to -601)	0.009

Table 4.1 detailing the estimated ATT and the resulting estimated change and percentage change in alcohol-attributable hospitalisations per year across Scotland.

4.2 Falsification Placebo Tests

Falsification placebo tests were undertaken by estimating the ATT as if MUP had been implemented on 1st May for each available year in the data, i.e. using every pretreatment two-year period as the Difference in Differences estimation period and checking if these placebos produced results which falsify the main estimate. Additionally, this process was repeated for every pretreatment one-year period. No placebo results were statistically significant to a chosen 0.05 threshold and therefore the placebo tests did not falsify the main ATT estimate, see Figure 4.2.1, Figure 4.2.2, Table 4.2.1, and Table 4.2.2.

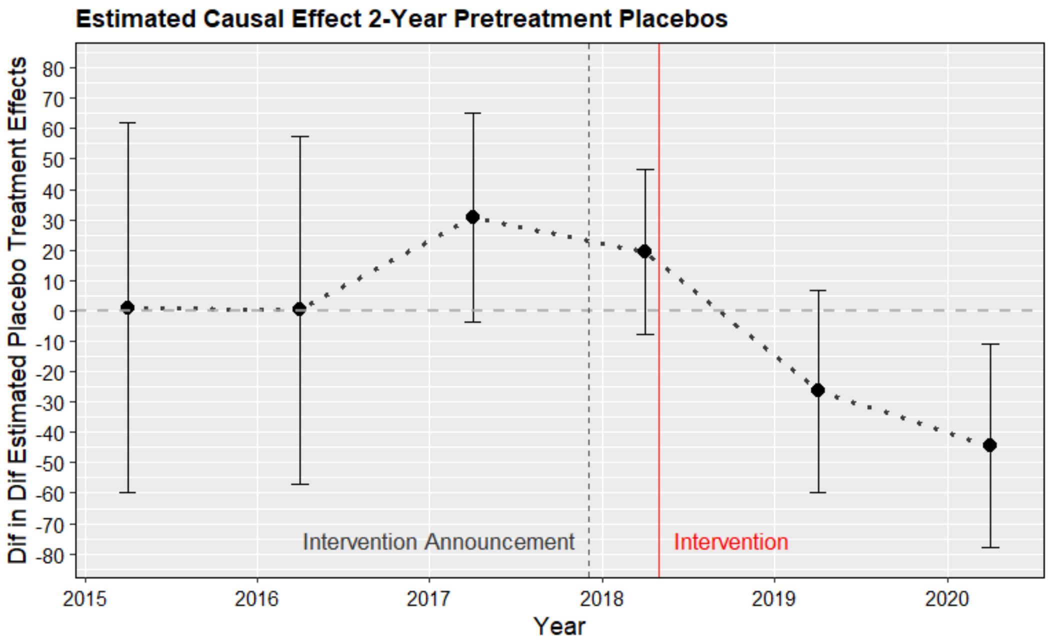

Figure 4.2.1 showing results for the two-year falsification placebo tests and the main ATT estimate, including 95% confidence intervals.

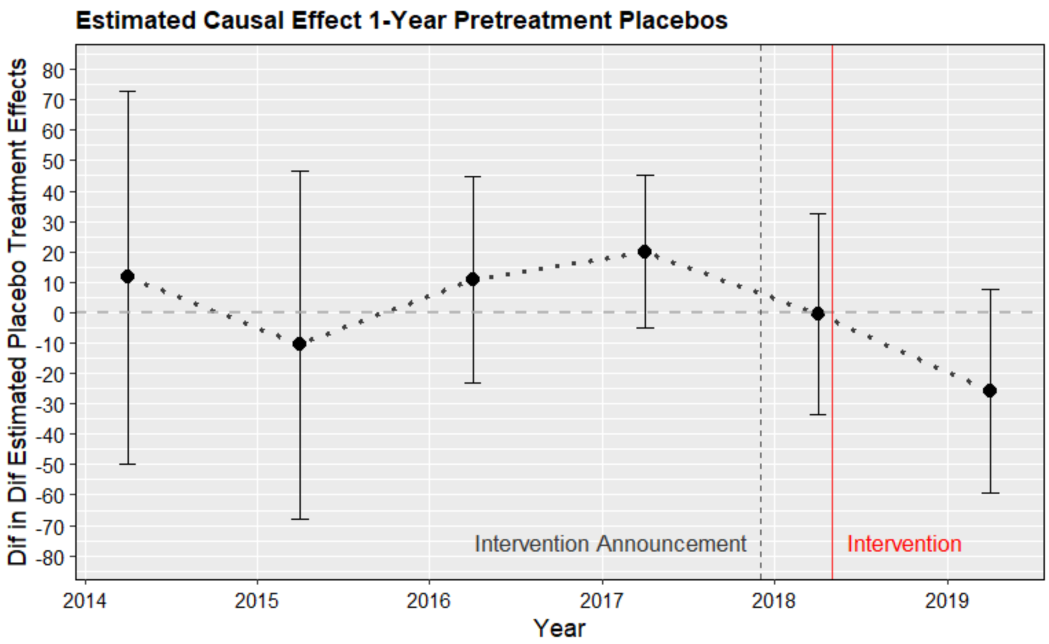

Figure 4.2.2 showing results for the one-year falsification placebo tests and a one-year ATT estimate, including 95% confidence intervals.

Financial Year	Estimated ATT	Lower 95% Confidence Interval	Upper 95% Confidence Interval	p-value
2014/15	1.01	-59.83	61.85	0.974
2015/16	0.30	-56.83	57.53	0.992
2016/17	30.86	-3.63	65.36	0.080
2017/18	19.40	-7.73	46.53	0.161
2018/19	-26.48	-59.83	6.88	0.120
2019/20	-44.52	-78.04	-11.00	0.009

Table 4.2.1 detailing results for all 2-year falsification placebo tests and the main Difference in Differences analysis (2019/2020).

Financial Year	Estimated ATT	Lower 95% Confidence Interval	Upper 95% Confidence Interval	p-value
2013/14	11.55	-49.71	72.80	0.712
2014/15	-10.54	-67.87	46.80	0.719
2015/16	10.84	-23.17	44.85	0.532
2016/17	20.03	-5.04	45.09	0.117
2017/18	-0.63	-33.69	32.43	0.970
2018/19	-25.85	-59.31	7.61	0.130

Table 4.2.2 detailing results for all 1-year falsification placebo tests and a 1-year Difference in Differences analysis (2018/2019).

4.3 Sensitivity Analysis

Sensitivity analysis was undertaken by estimating the ATT under a series of different methodological decisions to those of the main analysis to check if results are robust to these decisions.

Sensitivity Analyses:

A. Instead of MICE PMM imputation, simple mean imputation was used.
B. Instead of MICE PMM imputation, simple median imputation was used.
C. Instead of MICE PMM imputation, MICE linear regression imputation was used.
D. Instead of matching based on the variables detailed in Section 2.2, a series of ad hoc analyses were undertaken to assess if results were sensitive to the choice of matching variables. It was determined results were not sensitive, given the inclusion of at least one employment related variable, one income related variable, and one age related variable. Results reported are from matching on employment rate for 16- to 64-year-old residents, median annual gross income, and percentage of 16- to 64-year-old residents.

E. Instead of including all 32 Scottish Local Authorities, 30 were included with the two Local Authorities which border England, “Dumfries and Galloway” and “Scottish Borders” removed to test if potential one way noncompliance at the border might influence results.

The results of all sensitivity analyses indicated that the estimated ATT was robust to the modelling decisions made, see Table 4.3.

Sensitivity Analysis	**Estimated ATT**	**Lower 95% Confidence Interval**	**Upper 95% Confidence Interval**	**p-value**
A	-42.71	-75.95	-9.56	0.011
B	-45.52	-80.30	-10.00	0.012
C	-44.96	-79.73	-10.19	0.011
D	-44.52	-78.04	-11.00	0.009
E	-46.00	-82.00	-10.01	0.012
Main Estimate	-44.52	-78.04	-11.00	0.009

Table 4.3 detailing results for all sensitivity analyses and the main Difference in Differences analysis (Main Estimate).

5. Discussion

MUP in Scotland was estimated to have plausibly reduced mean Local Authority alcohol-attributable hospitalisations per 100,000 residents per year by 44.5 (95% CI -78.0 to -11.0; p-value = 0.009), representing an estimated 6.2% (95% CI -10.9% to -1.5%; p-value = 0.009) or approximately 2,450 (95% CI -4,300 to -600; p-value = 0.009) reduction in total alcohol-attributable hospitalisations per year across Scotland. These findings were similar in magnitude to the estimated 4.1% (95% CI -8.3% to 0.3%; p-value = 0.064) reduction in alcohol-attributable hospitalisations from Wyper et al, 2023. Adding a second causal estimate to the literature, using a different methodology and data, adds plausibility to the argument that the introduction of MUP reduced the level of alcohol-attributable health harms in Scotland, in line with the Scottish Governments MUP theory of change.

Limitations of this research included the lack of data availability necessitating a Difference in Differences methodology, and the 6.7% missing Local Authority data adding uncertainty to results. Additionally, given Local Authorities were matched based on only 2017 data, this would not capture any potential temporal trends in Local Authority characteristics.

Future extensions of this research could include applications to Public Health Scotland and NHS England to gain more granular data in terms of both temporal resolution and the unit of observation e.g. individuals, covering a longer total time period. This would facilitate the use of different causal inference techniques to potentially add additional plausibility to the causal relationship. Potentially more robust Local Authority matching could be undertaken using time series data to ensure temporal trends are accounted for when matching. Additionally, as the Scottish Government is increasing the MUP to 65p on 1st May 2024, this would provide an additional opportunity to estimate any potential causal effect of an increase in MUP.

6. Annex (Statistical Appendix)

Section A

Figure A.1 showing imputed values (red) compared to unimputed values (blue) for employment rate for 16- to 64-year-old residents, points jittered.

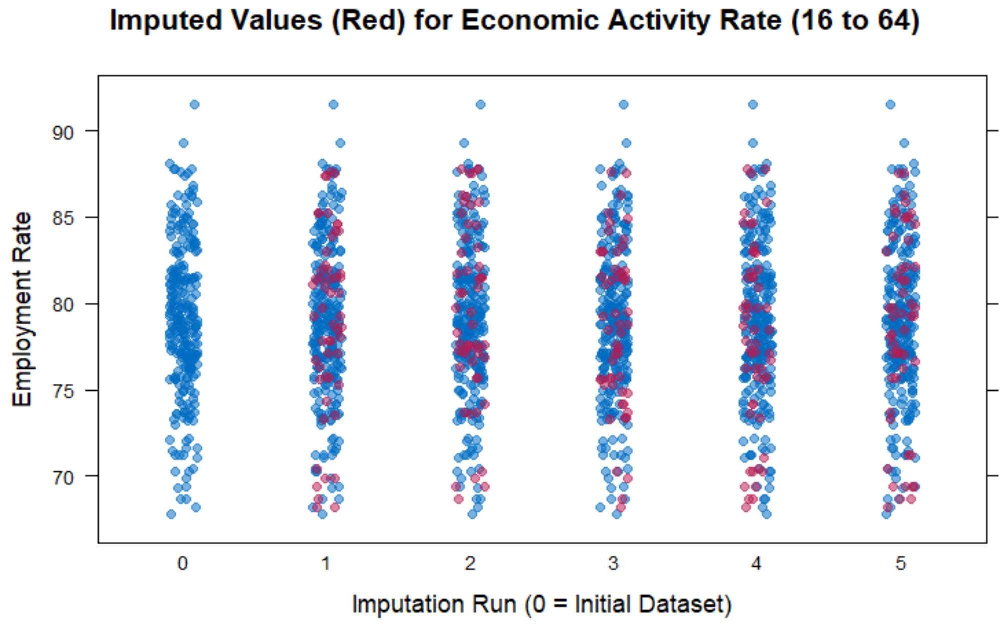

Figure A.2 showing imputed values (red) compared to unimputed values (blue) for economic activity rate for 16- to 64-year-old residents, points jittered.

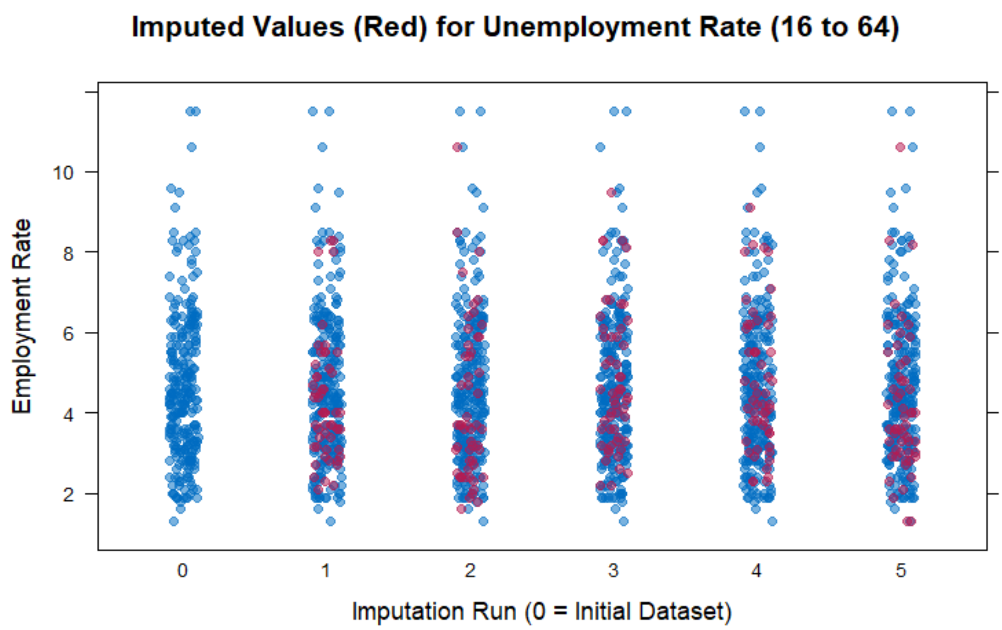

Figure A.3 showing imputed values (red) compared to unimputed values (blue) for unemployment rate for 16- to 64-year-old residents, points jittered.

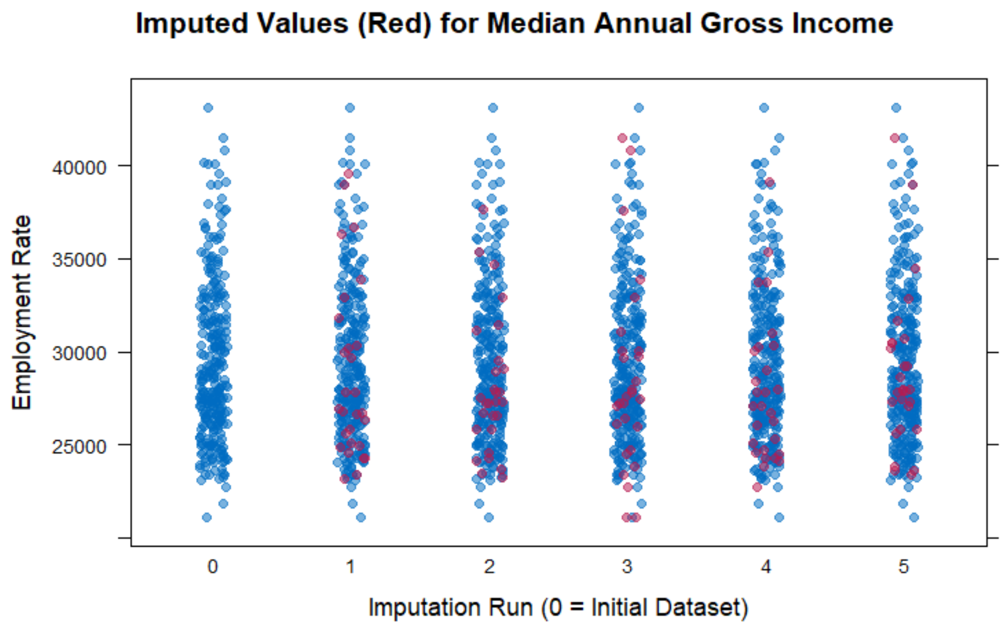

Figure A.4 showing imputed values (red) compared to unimputed values (blue) for median annual gross income, points jittered.

Section B

Scottish Local Authority	Matched English Local Authority
Aberdeen City	Brighton and Hove
Aberdeenshire	Wiltshire
Angus	North Kesteven
Argyll and Bute	Bassetlaw
City of Edinburgh	York
Clackmannanshire	County Durham
Dumfries and Galloway	Cornwall
Dundee City	West Lancashire
East Ayrshire	Sunderland
East Dunbartonshire	Broxbourne
East Lothian	Havering
East Renfrewshire	Tonbridge and Malling
Falkirk	Gateshead
Fife	Carlisle
Glasgow City	Liverpool
Highland	Fenland
Inverclyde	Oadby and Wigston
Midlothian	South Kesteven
Moray	Plymouth
Na h-Eileanan Siar	Allerdale
North Ayrshire	South Tyneside
North Lanarkshire	Leeds
Orkney Islands	Shropshire
Perth and Kinross	East Riding of Yorkshire
Renfrewshire	Canterbury
Scottish Borders	Northumberland
Shetland Islands	Rutland
South Ayrshire	Sefton
South Lanarkshire	Bath and North East Somerset
Stirling	Runnymede
West Dunbartonshire	Knowsley
West Lothian	Calderdale

Table B.1 detailing matched Scottish and English Local Authorities for imputation dataset 1.

Scottish Local Authority	Matched English Local Authority
Aberdeen City	Brighton and Hove
Aberdeenshire	Wiltshire
Angus	Craven
Argyll and Bute	Bassetlaw
City of Edinburgh	York
Clackmannanshire	Gateshead
Dumfries and Galloway	Cornwall
Dundee City	West Lancashire
East Ayrshire	County Durham
East Dunbartonshire	Tandridge
East Lothian	Havering
East Renfrewshire	Three Rivers
Falkirk	Plymouth
Fife	Bath and North East Somerset
Glasgow City	Liverpool
Highland	Harrogate
Inverclyde	Canterbury
Midlothian	Mid Sussex
Moray	Shropshire
Na h-Eileanan Siar	Mid Suffolk
North Ayrshire	South Tyneside
North Lanarkshire	Sunderland
Orkney Islands	Allerdale
Perth and Kinross	East Riding of Yorkshire
Renfrewshire	North Tyneside
Scottish Borders	Northumberland
Shetland Islands	Rutland
South Ayrshire	Sefton
South Lanarkshire	Southend-on-Sea
Stirling	Kingston upon Thames
West Dunbartonshire	Knowsley
West Lothian	Calderdale

Table B.2 detailing matched Scottish and English Local Authorities for imputation dataset 2.

Scottish Local Authority	Matched English Local Authority
Aberdeen City	Brighton and Hove
Aberdeenshire	Central Bedfordshire
Angus	County Durham
Argyll and Bute	Bassetlaw
City of Edinburgh	York
Clackmannanshire	Gateshead
Dumfries and Galloway	Cornwall
Dundee City	West Lancashire
East Ayrshire	South Tyneside
East Dunbartonshire	North Kesteven
East Lothian	Havering
East Renfrewshire	Barnet
Falkirk	Plymouth
Fife	Oadby and Wigston
Glasgow City	Liverpool
Highland	Fenland
Inverclyde	Sunderland
Midlothian	Mid Sussex
Moray	North Lincolnshire
Na h-Eileanan Siar	Mid Suffolk
North Ayrshire	Cotswold
North Lanarkshire	Leeds
Orkney Islands	Shropshire
Perth and Kinross	East Riding of Yorkshire
Renfrewshire	Canterbury
Scottish Borders	Northumberland
Shetland Islands	Rutland
South Ayrshire	Sefton
South Lanarkshire	Bath and North East Somerset
Stirling	Kingston upon Thames
West Dunbartonshire	Knowsley
West Lothian	Calderdale

Table B.3 detailing matched Scottish and English Local Authorities for imputation dataset 3.

Scottish Local Authority	**Matched English Local Authority**
Aberdeen City	Brighton and Hove
Aberdeenshire	Wiltshire
Angus	North Kesteven
Argyll and Bute	Bassetlaw
City of Edinburgh	York
Clackmannanshire	County Durham
Dumfries and Galloway	Cornwall
Dundee City	West Lancashire
East Ayrshire	South Tyneside
East Dunbartonshire	Castle Point
East Lothian	Havering
East Renfrewshire	Tonbridge and Malling
Falkirk	Gateshead
Fife	Bath and North East Somerset
Glasgow City	Liverpool
Highland	Harrogate
Inverclyde	Oadby and Wigston
Midlothian	Mid Sussex
Moray	Plymouth
Na h-Eileanan Siar	North East Derbyshire
North Ayrshire	Winchester
North Lanarkshire	Sunderland
Orkney Islands	Shropshire
Perth and Kinross	East Riding of Yorkshire
Renfrewshire	Runnymede
Scottish Borders	Northumberland
Shetland Islands	Rutland
South Ayrshire	Sefton
South Lanarkshire	Southend-on-Sea
Stirling	Kingston upon Thames
West Dunbartonshire	Knowsley
West Lothian	Calderdale

Table B.4 detailing matched Scottish and English Local Authorities for imputation dataset 4.

Scottish Local Authority	**Matched English Local Authority**
Aberdeen City	Brighton and Hove
Aberdeenshire	Central Bedfordshire
Angus	Craven
Argyll and Bute	Bassetlaw
City of Edinburgh	York
Clackmannanshire	Gateshead
Dumfries and Galloway	Cornwall
Dundee City	West Lancashire
East Ayrshire	County Durham
East Dunbartonshire	Broxbourne
East Lothian	Havering
East Renfrewshire	Barnet
Falkirk	Plymouth
Fife	North East Derbyshire
Glasgow City	Liverpool
Highland	North Warwickshire
Inverclyde	Canterbury
Midlothian	South Kesteven
Moray	Shropshire
Na h-Eileanan Siar	Allerdale
North Ayrshire	South Tyneside
North Lanarkshire	Sunderland
Orkney Islands	Barrow-in-Furness
Perth and Kinross	East Riding of Yorkshire
Renfrewshire	Runnymede
Scottish Borders	Northumberland
Shetland Islands	Rutland
South Ayrshire	Sefton
South Lanarkshire	Bath and North East Somerset
Stirling	Kingston upon Thames
West Dunbartonshire	Knowsley
West Lothian	Calderdale

Table B.5 detailing matched Scottish and English Local Authorities for imputation dataset 5.

Section C

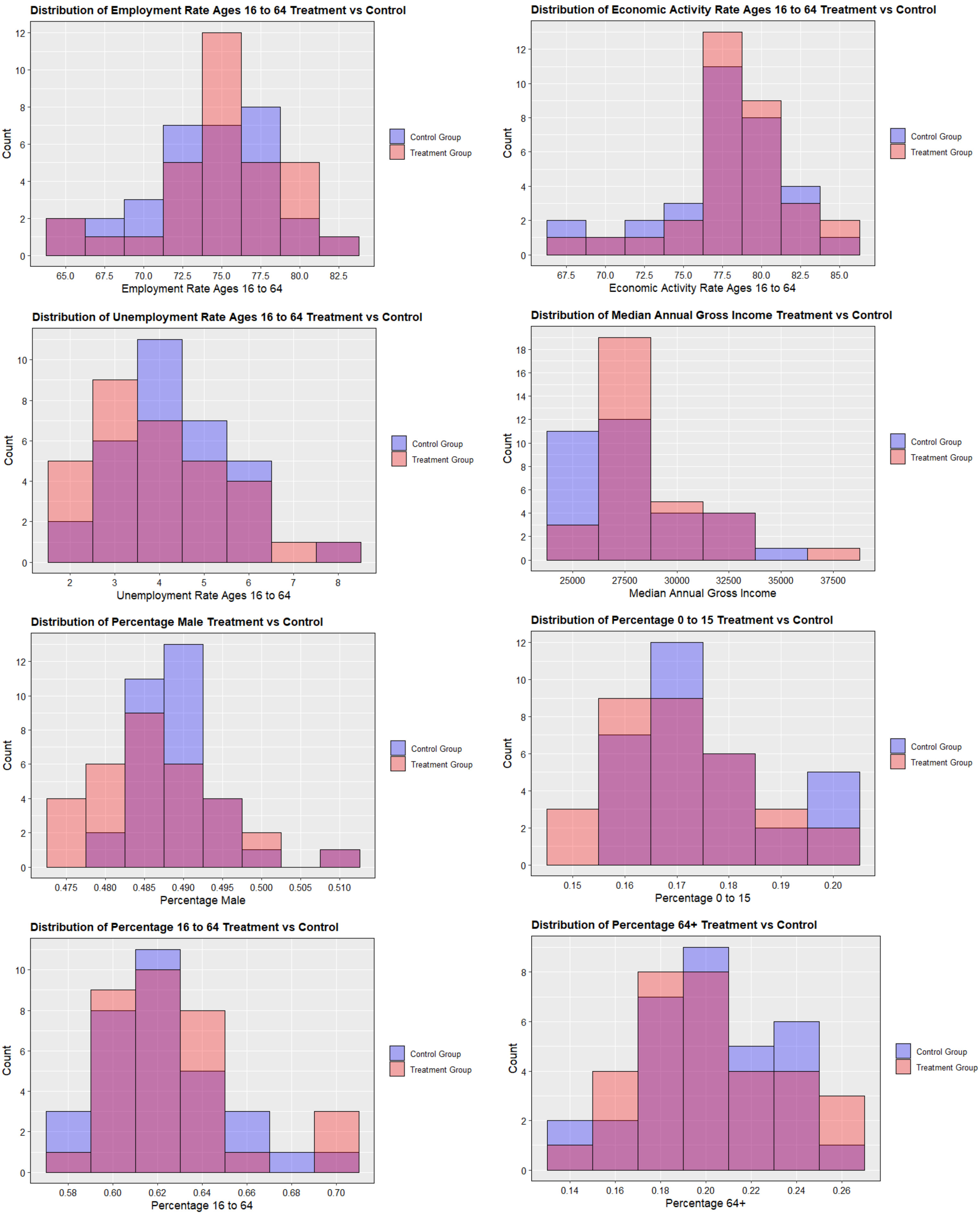

Figure C.1 showing the distribution of each matching variable between control and treatment groups, imputed dataset 1.

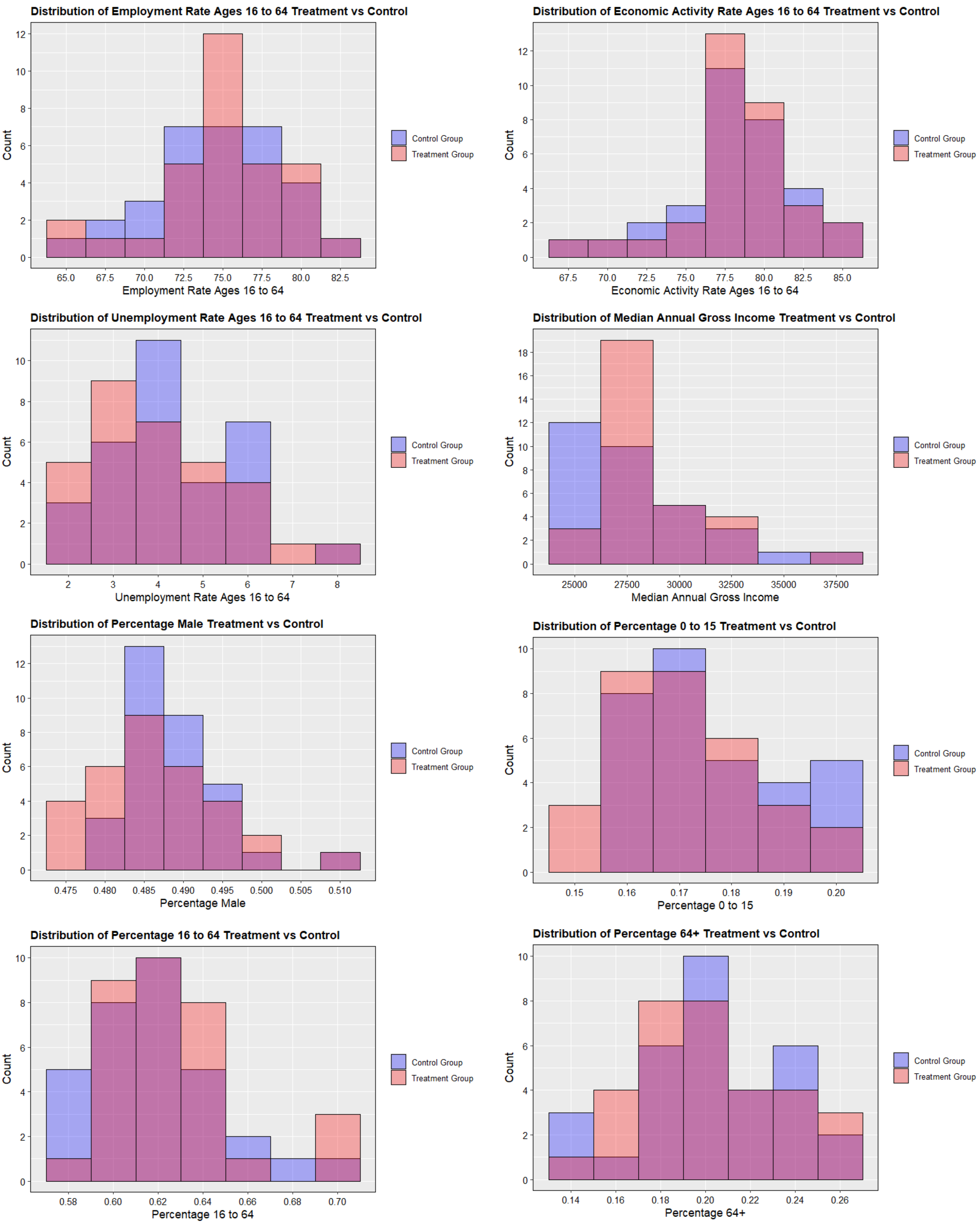

Figure C.2 showing the distribution of each matching variable between control and treatment groups, imputed dataset 2.

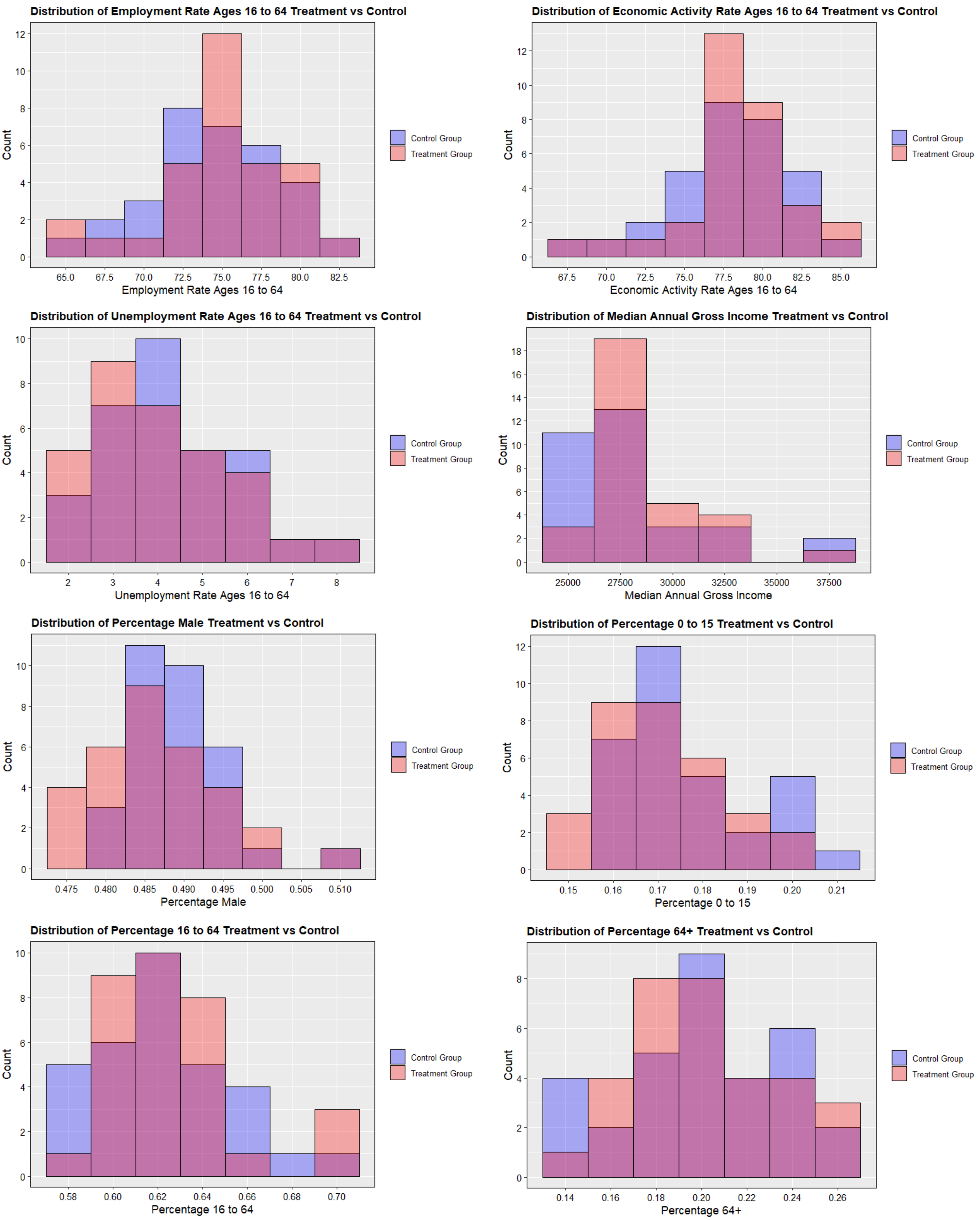

Figure C.3 showing the distribution of each matching variable between control and treatment groups, imputed dataset 3.

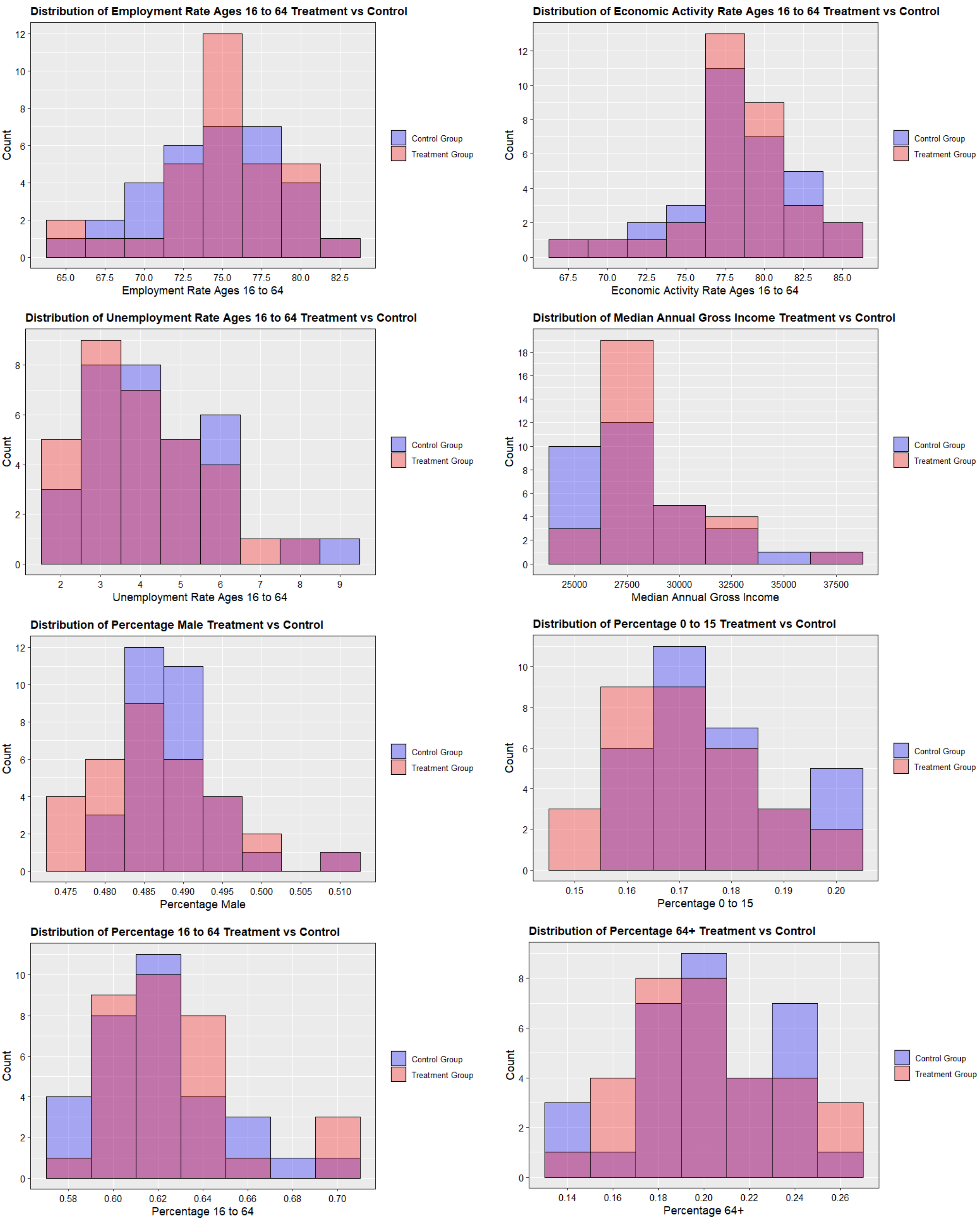

Figure C.4 showing the distribution of each matching variable between control and treatment groups, imputed dataset 4.

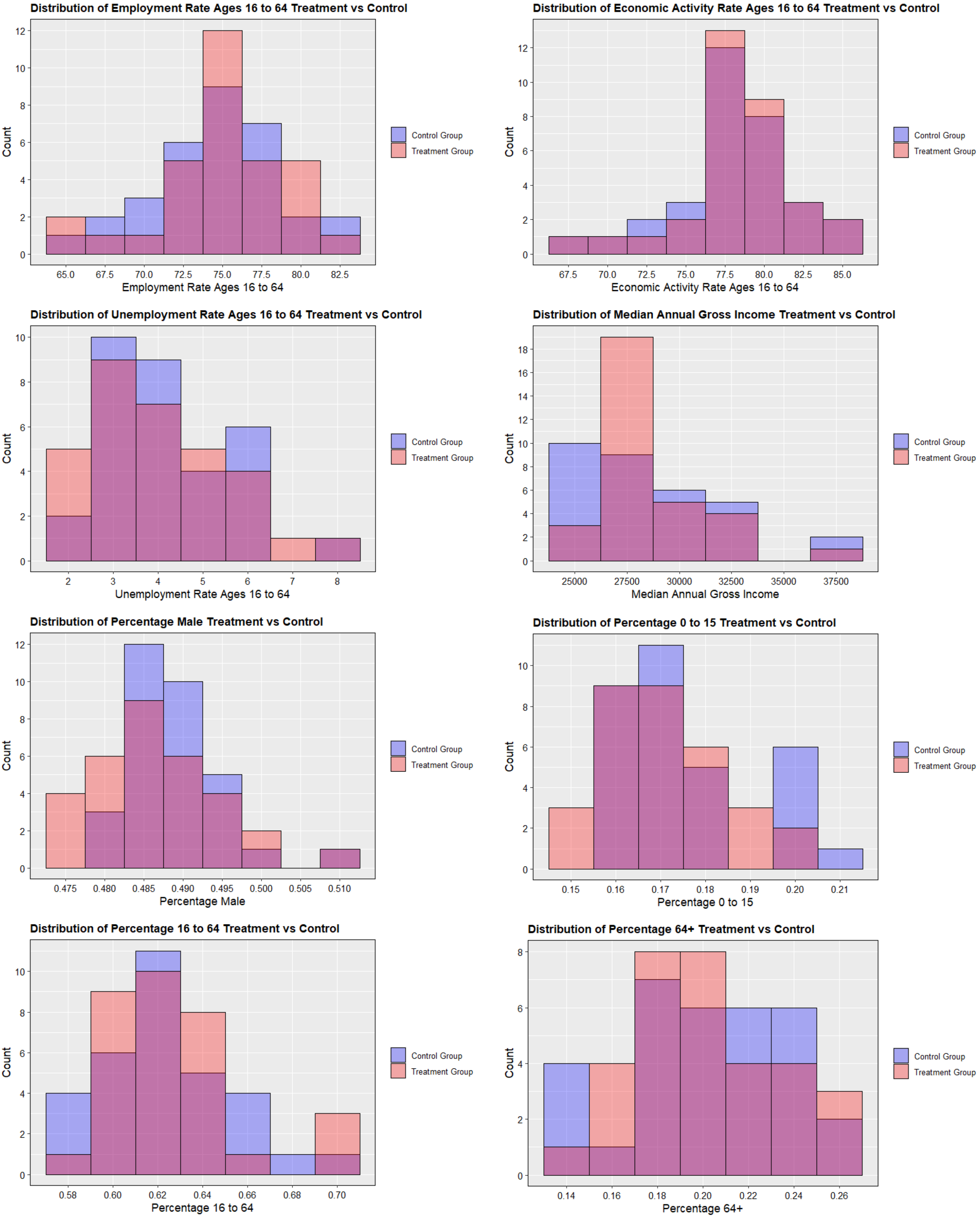

Figure C.5 showing the distribution of each matching variable between control and treatment groups, imputed dataset 5.

Section D

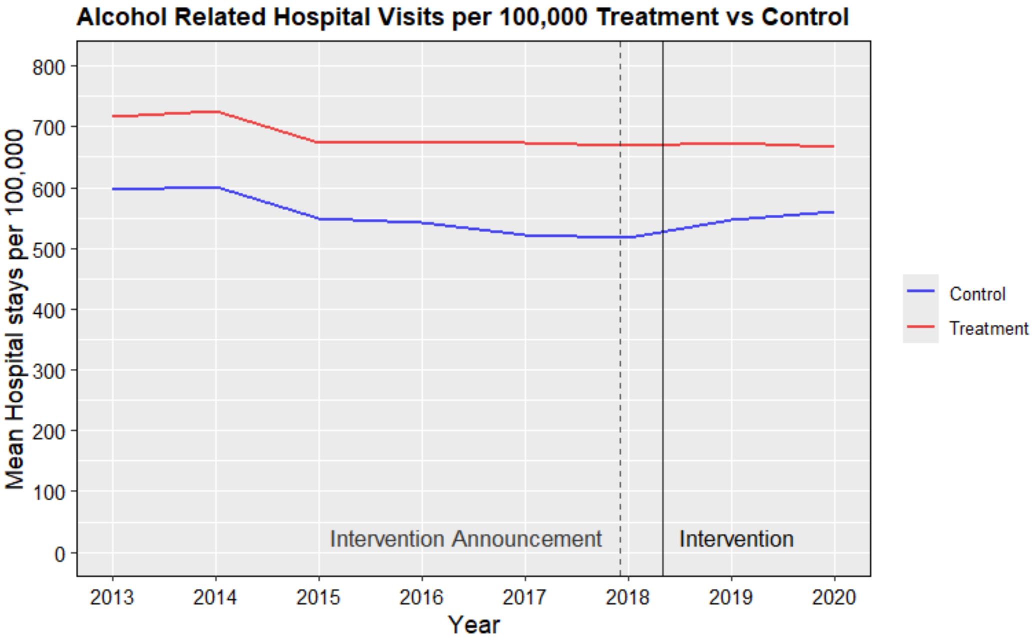

Figure D.1 showing the mean Local Authority alcohol-attributable hospitalisations per 100,000 residents per year for the treatment and control group for imputed dataset 1.

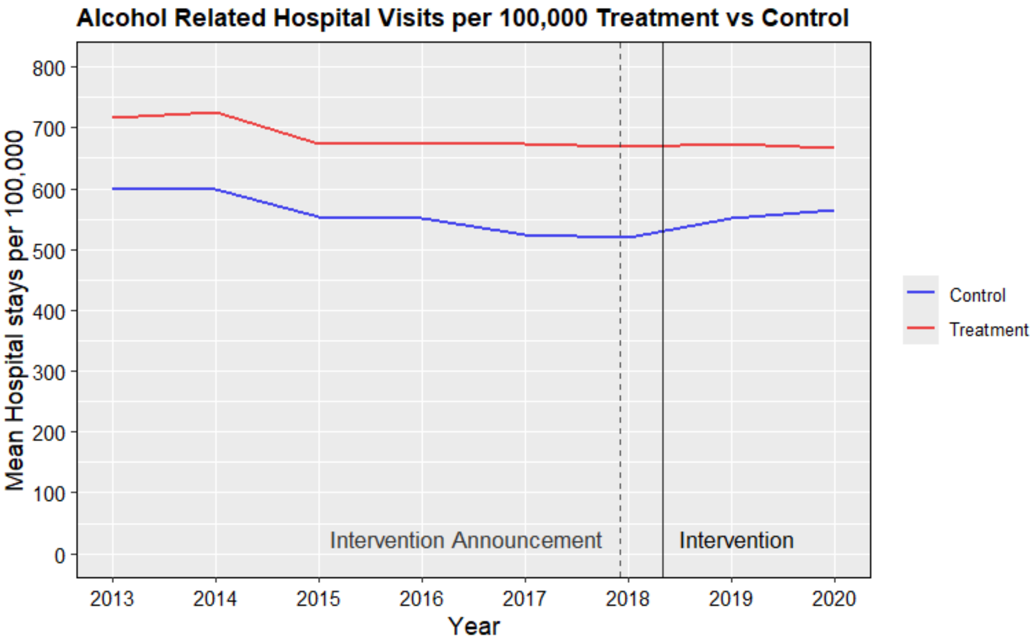

Figure D.2 showing the mean Local Authority alcohol-attributable hospitalisations per 100,000 residents per year for the treatment and control group for imputed dataset 2.

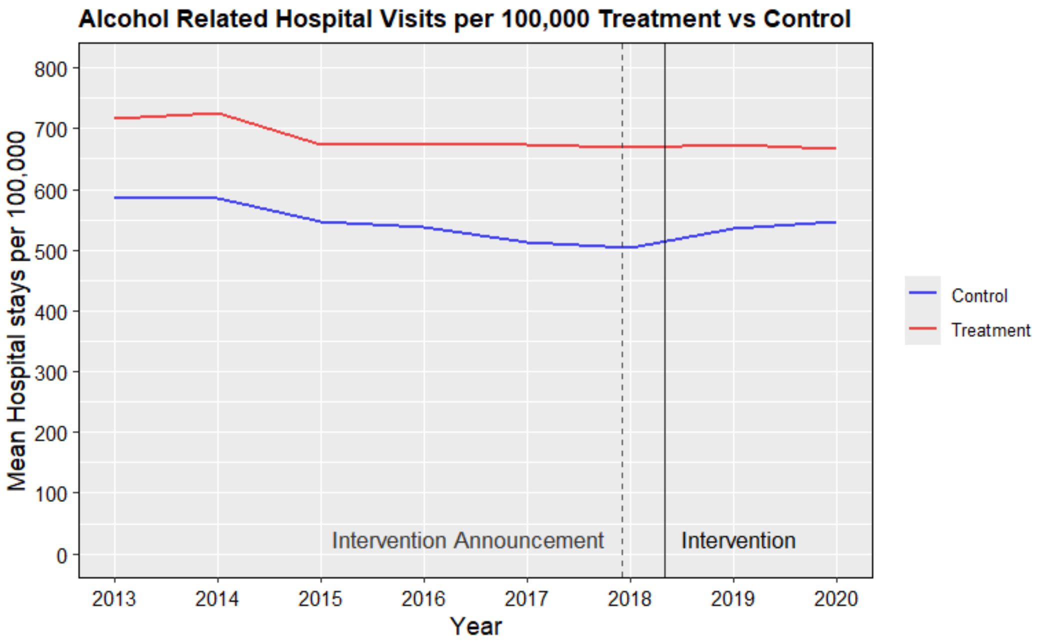

Figure D.3 showing the mean Local Authority alcohol-attributable hospitalisations per 100,000 residents per year for the treatment and control group for imputed dataset 3.

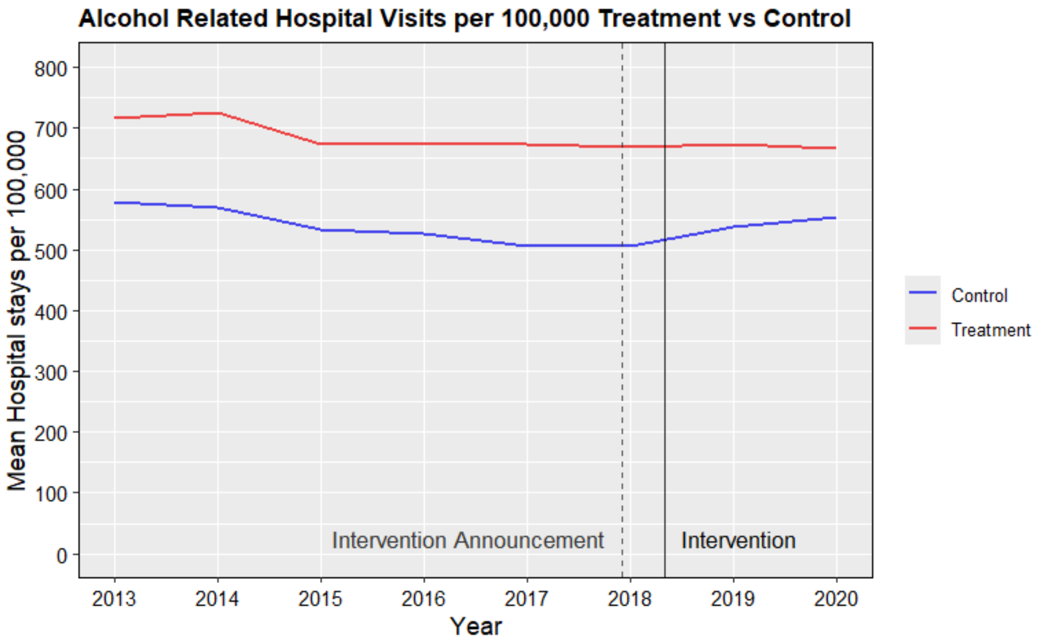

Figure D.4 showing the mean Local Authority alcohol-attributable hospitalisations per 100,000 residents per year for the treatment and control group for imputed dataset 4.

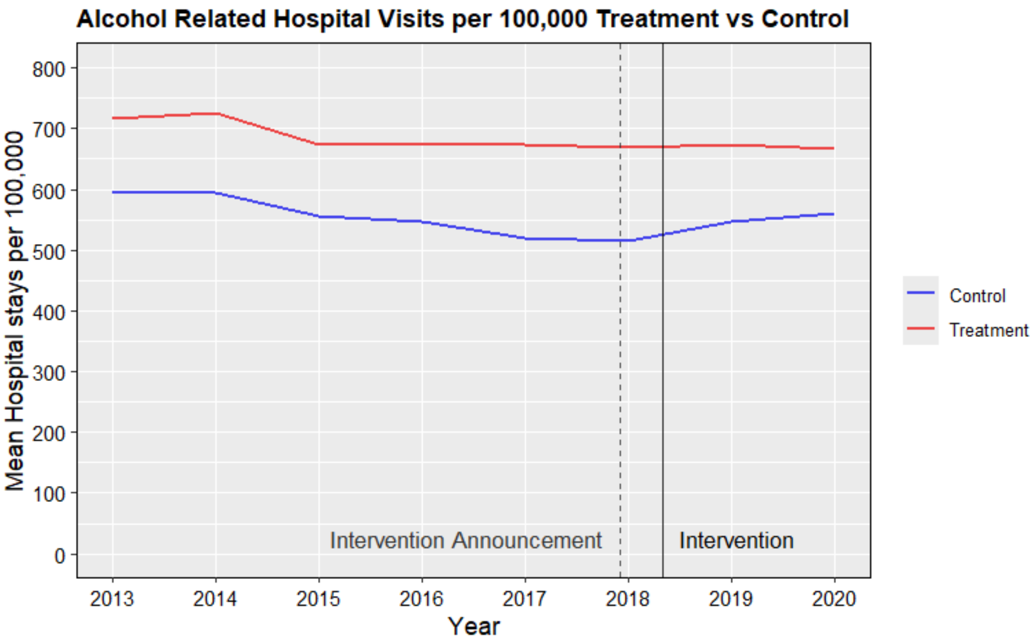

Figure D.5 showing the mean Local Authority alcohol-attributable hospitalisations per 100,000 residents per year for the treatment and control group for imputed dataset 5.

7. Bibliography

Austin, P. C., 2011. An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies. Multivariate Behavioral Research, 46(3), 399-424. DOI: 10.1080/00273171.2011.568786

Azur, M., Stuart, E., Frangakis, C., & Leaf, P., 2011. Multiple imputation by chained equations: what is it and how does it work?. International Journal of Methods in Psychiatric Research, 20. https://doi.org/10.1002/mpr.329.

Brenner, A., Borrell, L., Barrientos-Gutiérrez, T., & Roux, A., 2015. Longitudinal associations of neighborhood socioeconomic characteristics and alcohol availability on drinking: Results from the Multi-Ethnic Study of Atherosclerosis (MESA). Social science & medicine, 145, pp. 17-25. https://doi.org/10.1016/j.socscimed.2015.09.030.

CRAN., 2024. mice: Multivariate Imputation by Chained Equations. https://cran.r-project.org/package=mice

De Maesschalck, R., Jouan-Rimbaud, D., & Massart, D. L., 2000. The Mahalanobis distance. Chemometrics and Intelligent Laboratory Systems, 50(1), 1-18. DOI: 10.1016/S0169-7439(99)00047-7

Efron, B., & Tibshirani, R., 1986. Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy. Statistical Science, 1, pp. 54-75. https://doi.org/10.1214/SS/1177013815.

Giles, L., Mackay, D., Richardson, E., Lewsey, J., Beeston, C., and Robinson, M., 2022. Evaluating the impact of minimum unit pricing (MUP) on sales-based alcohol consumption in Scotland at three years post-implementation.

Imbens, G., & Rubin, D., 2011. Rubin Causal Model. pp. 1263-1265. https://doi.org/10.1007/978-3-642-04898-2_64.

Jones, L., Bates, G., McCoy, E., & Bellis, M., 2015. Relationship between alcohol-attributable disease and socioeconomic status, and the role of alcohol consumption in this relationship: a systematic review and meta-analysis. BMC Public Health, 15. https://doi.org/10.1186/s12889-015-1720-7.

Khamis, A., Salleh, S., Karim, M., Rom, N., Janasekaran, S., Idris, A., & Rashid, R., 2022. Alcohol Consumption Patterns: A Systematic Review of Demographic and Sociocultural Influencing Factors. International Journal of Environmental Research and Public Health, 19. https://doi.org/10.3390/ijerph19138103.

Mahalanobis, P.C., 1930. On tests and measures of groups divergence. Journal of Asiatic Sociology of Bengal, 26 (1930), pp. 541-588.

National Records of Scotland, NRS., 2024. https://www.nrscotland.gov.uk/statistics-and-data/statistics/statistics-by-theme/vital-events/deaths

NHS Digital, NHS., 2024. Hospital Episode Statistics. https://digital.nhs.uk/data-and-information/data-tools-and-services/data-services/hospital-episode-statistics

Probst, C., Roerecke, M., Behrendt, S., & Rehm, J., 2014. Socioeconomic differences in alcohol-attributable mortality compared with all-cause mortality: a systematic review and meta-analysis. International journal of epidemiology, 43 4, pp. 1314-27. https://doi.org/10.1093/ije/dyu043.

Public Health Scotland, PHS., 2018. Minimum unit pricing (MUP) theory of change. http://www.healthscotland.scot/publications/minimum-unit-pricing-mup-theory-of-change

Public Health Scotland, PHS., 2024. SMR datasets. https://publichealthscotland.scot/services/national-data-catalogue/

Rubin, D. B., 1987. Multiple Imputation for Nonresponse in Surveys. John Wiley & Sons.

Scottish Government. 2016. Scottish Index of Multiple Deprivation. https://www.gov.scot/publications/scottish-index-multiple-deprivation-2016/

Stockwell, T., Auld, M., Zhao, J., & Martin, G., 2012. Does minimum pricing reduce alcohol consumption? The experience of a Canadian province. Addiction, 107 5, pp. 912-20. https://doi.org/10.1111/j.1360-0443.2011.03763.x.

Stockwell, T., Zhao, J., Martin, G., Macdonald, S., Vallance, K., Treno, A., Ponicki, W., Tu, A., & Buxton, J., 2013. Minimum alcohol prices and outlet densities in British Columbia, Canada: estimated impacts on alcohol-attributable hospital admissions. American journal of public health, 103 11, pp. 2014-20 . https://doi.org/10.2105/AJPH.2013.301289.

Temple, M., Fillmore, K., Hartka, E., Johnstone, B., Leino, E., & Motoyoshi, M., 1991. A meta-analysis of change in marital and employment status as predictors of alcohol consumption on a typical occasion. British journal of addiction, 86 10, pp. 1269-81. https://doi.org/10.1111/J.1360-0443.1991.TB01703.X.

UK Government. 2010. English indices of deprivation 2010. https://www.gov.uk/government/statistics/english-indices-of-deprivation-2010

Wilson, L., Pryce, R., Angus, C., Hiscock, R., Breennan, A., and Gillespie, D., 2021. The effect of alcohol tax changes on retail prices: how do on-trade alcohol retailers pass through tax changes to consumers?. Eur J Health Econ. 2021; 22: 381-392

Wyper, G., Mackay, D., Fraser, C., Lewsey, J., Robinson, M., Beeston, C., and Giles, L., 2023. Evaluating the impact of alcohol minimum unit pricing on deaths and hospitalisations in Scotland: a controlled interrupted time series study

Zhao, J., & Stockwell, T., 2017. The impacts of minimum alcohol pricing on alcohol attributable morbidity in regions of British Colombia, Canada with low, medium and high mean family income. Addiction, 112, pp. 1942–1951. https://doi.org/10.1111/add.13902.

Zhao, J., Stockwell, T., Martin, G., Macdonald, S., Vallance, K., Treno, A., Ponicki, W., Tu, A., & Buxton, J., 2013. The relationship between minimum alcohol prices, outlet densities and alcohol-attributable deaths in British Columbia, 2002-09. Addiction, 108 6, pp. 1059-69. https://doi.org/10.1111/add.12139.

Zhao, Z., 2004. Using matching to estimate treatment effects: data requirements, matching metrics, and Monte Carlo evidence. The Review of Economics and Statistics, 86 (1) (2004), pp. 91-107.

Zio, M., & Guarnera, U., 2009. Semiparametric predictive mean matching. AStA Advances in Statistical Analysis, 93, 175-186. https://doi.org/10.1007/S10182-008-0081-2.

8. Code Annex

````
---
title: "CI_Assignment"
output: html_document
---

```{r setup, include = FALSE}

knitr::opts_chunk$set(echo = TRUE)

```

## Packages

```{r include = FALSE}

library(tidyverse)
library(ggplot2)
library(scales)
library(gridExtra)
library(corrplot)
library(Hmisc)
library(mice)
library(MatchIt)
library(boot)

```

## Flags

```{r}

# Imputation flags
# If all FALSE removes any rows with NAs as opposed to imputing
mean_imputation = FALSE
median_imputation = FALSE
mice_imputation = TRUE
n_mice_imputations = 5

```

## Initial Data Loading and Checks

```{r}

# Load data

# Scotland alcohol related hospital visits by local authority
filepath = "scotland_hospital_visits_local_authority.csv"
scotland_hospital_visits_local_authority = read.csv(filepath, header = TRUE)

# Scotland alcohol related hospital visits national
filepath = "scotland_hospital_visits_whole.csv"
scotland_hospital_visits_whole = read.csv(filepath, header = TRUE)

# Scotland population by local authority and national
filepath = "scotland_population_local_authority_and_whole.csv"
scotland_population_local_authority_whole = read.csv(filepath, header = TRUE)

# England alcohol related hospital visits per 100k by local authority
filepath = "england_hospital_visits_per100k_local_authority_joined.csv"
england_hospital_visits_local_authority = read.csv(filepath, header = TRUE)

# UK annual population survey data by local authority - 2017
filepath = "annual_population_survey.csv"
uk_pop_survey_2017 = read.csv(filepath, header = TRUE)
````
````

````
# UK annual hours and earnings data by local authority - 2017
filepath = "annual_hours_and_earnings_survey.csv"
uk_earnings_survey_2017 = read.csv(filepath, header = TRUE)

# UK annual population survey data by age, sex, local authority - 2017
filepath = "annual_population_age_sex_survey.csv"
uk_pop_ages_sex_survey_2017 = read.csv(filepath, header = TRUE)

```

```{r}

# Preprocess data

# Scotland alcohol related hospital visits by local authority
scotland_hospital_visits_local_authority_clean = scotland_hospital_visits_local_authority %>%
  dplyr::select(financial_year, local_authority, sg_code, stays) %>%
  dplyr::mutate(year = str_sub(str_trim(financial_year), start = 1, end = 4)) %>%
  dplyr::mutate(year = as.numeric(year)) %>%
  dplyr::mutate(local_authority = dplyr::case_when(local_authority == "Edinburgh City" ~ "City of
Edinburgh",
                   local_authority == "Ayrshire East" ~ "East Ayrshire",
                   local_authority == "Dunbartonshire East" ~ "East Dunbartonshire",
                   local_authority == "Lothian East" ~ "East Lothian",
                   local_authority == "Renfrewshire East" ~ "East Renfrewshire",
                   local_authority == "Ayrshire North" ~ "North Ayrshire",
                   local_authority == "Lanarkshire North" ~ "North Lanarkshire",
                   local_authority == "Borders" ~ "Scottish Borders",
                   local_authority == "Ayrshire South" ~ "South Ayrshire",
                   local_authority == "Lanarkshire South" ~ "South Lanarkshire",
                   local_authority == "Dunbartonshire West" ~ "West Dunbartonshire",
                   local_authority == "Lothian West" ~ "West Lothian",
                   TRUE ~ local_authority))

# Check for NAs
colSums(is.na(scotland_hospital_visits_local_authority_clean))

# Scotland alcohol related hospital visits national
scotland_hospital_visits_whole_clean = scotland_hospital_visits_whole %>%
  dplyr::select(financial_year, grouping, stays) %>%
  dplyr::mutate(year = str_sub(str_trim(financial_year), start = 1, end = 4)) %>%
  dplyr::mutate(year = as.numeric(year))

# Check for NAs
colSums(is.na(scotland_hospital_visits_whole_clean))

# Scotland yearly population by local authority
scotland_population_local_authority_clean = scotland_population_local_authority_whole %>%
  dplyr::filter(local_authority != "Scotland") %>%
  dplyr::mutate(year = as.numeric(year))

# Check for NAs
colSums(is.na(scotland_population_local_authority_clean))

# Check local authority names match
sort(unique(scotland_hospital_visits_local_authority_clean$local_authority)) ==
sort(unique(scotland_population_local_authority_clean$local_authority))

# Scotland yearly population national
scotland_population_whole_clean = scotland_population_local_authority_whole %>%
  dplyr::filter(local_authority == "Scotland")

# Create Scotland alcohol related hospital visits per 100,00 by local authority
scotland_hospital_visits_local_authority_per100k = scotland_hospital_visits_local_authority_clean %>%
  dplyr::left_join(scotland_population_local_authority_clean, by = c("local_authority", "year")) %>%
  dplyr::mutate(stays_per100k = (stays * 100000) / population) # %>%
  # dplyr::filter(local_authority != "Dumfries and Galloway" & local_authority != "Scottish Borders")
# Remove border local authorities check
````

```
# Create Scotland alcohol related hospital visits per 100,00 national
scotland_hospital_visits_whole_per100k = scotland_hospital_visits_whole_clean %>%
  dplyr::left_join(scotland_population_whole_clean, by = c("year")) %>%
  dplyr::mutate(stays_per100k = (stays * 100000) / population) %>%
  dplyr::filter(year >= 1997)

# England alcohol related hospital visits by local authority
england_hospital_visits_local_authority_per100k = england_hospital_visits_local_authority %>%
  dplyr::mutate(year = str_sub(str_trim(financial_year), start = 1, end = 4)) %>%
  dplyr::select(financial_year, year, local_authority, stays_per100k) %>%
  dplyr::mutate(stays_per100k = as.numeric(stays_per100k),
                year = as.numeric(year)) %>%
  dplyr::filter(!is.na(stays_per100k)) %>%
  dplyr::group_by(local_authority) %>%
  dplyr::filter(n() >= 8) %>%
  dplyr::ungroup()

# UK annual population survey data by local authority - 2017
uk_pop_survey_2017_clean = uk_pop_survey_2017 %>%
  dplyr::mutate(economic_activity_rate_16to64 = as.numeric(economic_activity_rate_16to64),
                employment_rate_16to64 = as.numeric(employment_rate_16to64),
                unemployment_rate_16to64 = as.numeric(unemployment_rate_16to64)) %>%
  dplyr::select(-unemployment_rate_16plus) %>%
  dplyr::filter_all(all_vars(!is.na(.)))

# UK annual hours and earnings data by local authority - 2017
uk_earnings_survey_2017_clean = uk_earnings_survey_2017 %>%
  dplyr::mutate(median_weekly_pay_gross = as.numeric(median_weekly_pay_gross),
                median_annual_pay_gross = as.numeric(median_annual_pay_gross),
                mean_weekly_pay_gross = as.numeric(mean_weekly_pay_gross),
                mean_annual_pay_gross = as.numeric(mean_annual_pay_gross)) %>%
  dplyr::filter_all(all_vars(!is.na(.)))

# UK annual population survey data by age, sex, local authority - 2017
uk_pop_ages_sex_survey_2017_clean = uk_pop_ages_sex_survey_2017 %>%
  dplyr::mutate(pop_all = as.numeric(pop_all),
                pop_0to15 = as.numeric(pop_0to15),
                pop_16to64 = as.numeric(pop_16to64),
                pop_65plus = as.numeric(pop_65plus),
                pop_all_m = as.numeric(pop_all_m),
                pop_0to15_m = as.numeric(pop_0to15_m),
                pop_16to64_m = as.numeric(pop_16to64_m),
                pop_65plus_m = as.numeric(pop_65plus_m),
                pop_all_f = as.numeric(pop_all_f),
                pop_0to15_f = as.numeric(pop_0to15_f),
                pop_16to64_f = as.numeric(pop_16to64_f),
                pop_65plus_f = as.numeric(pop_65plus_f)) %>%
  dplyr::filter_all(all_vars(!is.na(.))) %>%
  dplyr::mutate(percentage_male = pop_all_m / (pop_all_m + pop_all_f),
                percentage_0to15 = pop_0to15 / pop_all,
                percentage_16to64 = pop_16to64 / pop_all,
                percentage_65plus = pop_65plus / pop_all)

# Create local authority matching dataset
local_authority_characteristics = uk_pop_survey_2017_clean %>%
  dplyr::inner_join(uk_earnings_survey_2017_clean, by = c("local_authority", "area_code")) %>%
  dplyr::inner_join(uk_pop_ages_sex_survey_2017_clean, by = c("local_authority", "area_code"))

if (mean_imputation == TRUE) {

  # Create local authority matching dataset
  local_authority_characteristics = uk_pop_survey_2017_clean %>%
    dplyr::full_join(uk_earnings_survey_2017_clean, by = c("local_authority", "area_code")) %>%
    dplyr::full_join(uk_pop_ages_sex_survey_2017_clean, by = c("local_authority", "area_code")) %>%
    dplyr::filter(local_authority %in%
unique(england_hospital_visits_local_authority_per100k$local_authority) | local_authority %in%
unique(scotland_hospital_visits_local_authority_per100k$local_authority)) %>%
    dplyr::select(c(local_authority, economic_activity_rate_16to64, employment_rate_16to64,
unemployment_rate_16to64,
```

```
                  median_annual_pay_gross, percentage_male, percentage_0to15, percentage_16to64,
percentage_65plus))

  local_authority_characteristics = data.frame(lapply(local_authority_characteristics, function(x)
impute(x, mean)))

}

if (median_imputation == TRUE) {

  # Create local authority matching dataset
  local_authority_characteristics = uk_pop_survey_2017_clean %>%
    dplyr::full_join(uk_earnings_survey_2017_clean, by = c("local_authority", "area_code")) %>%
    dplyr::full_join(uk_pop_ages_sex_survey_2017_clean, by = c("local_authority", "area_code")) %>%
    dplyr::filter(local_authority %in%
unique(england_hospital_visits_local_authority_per100k$local_authority) | local_authority %in%
unique(scotland_hospital_visits_local_authority_per100k$local_authority)) %>%
    dplyr::select(c(local_authority, economic_activity_rate_16to64, employment_rate_16to64,
unemployment_rate_16to64,
                  median_annual_pay_gross, percentage_male, percentage_0to15, percentage_16to64,
percentage_65plus))

  local_authority_characteristics = data.frame(lapply(local_authority_characteristics, function(x)
impute(x, median)))

}

if (mice_imputation == TRUE) {

  # Create local authority matching dataset
  local_authority_characteristics = uk_pop_survey_2017_clean %>%
    dplyr::full_join(uk_earnings_survey_2017_clean, by = c("local_authority", "area_code")) %>%
    dplyr::full_join(uk_pop_ages_sex_survey_2017_clean, by = c("local_authority", "area_code")) %>%
    dplyr::filter(local_authority %in%
unique(england_hospital_visits_local_authority_per100k$local_authority) | local_authority %in%
unique(scotland_hospital_visits_local_authority_per100k$local_authority)) %>%
    dplyr::select(c(local_authority, economic_activity_rate_16to64, employment_rate_16to64,
unemployment_rate_16to64,
                  median_annual_pay_gross, percentage_male, percentage_0to15, percentage_16to64,
percentage_65plus))

  # Use MICE to impute missing values and produce 5 datasets with different imputed missing values
  pred_matrix = quickpred(local_authority_characteristics)

  pred_matrix[, "local_authority"] = 0

  methods = rep("pmm", ncol(local_authority_characteristics))
  names(methods) = colnames(local_authority_characteristics)
  methods["local_authority"] = ""

  # local_authority_characteristics = mice(local_authority_characteristics, m = 5, method = methods,
predictorMatrix = pred_matrix, seed = 12345, printFlag = TRUE)

  mice_model = mice(local_authority_characteristics, m = 5, method = methods, predictorMatrix =
pred_matrix, seed = 12345, printFlag = TRUE)

  local_authority_characteristics_list = list()

  for(i in 1:n_mice_imputations) {

    local_authority_characteristics_list[[i]] = complete(mice_model, action = i)

  }


}

england_hospital_visits_local_authority_per100k_has_characteristics =
england_hospital_visits_local_authority_per100k %>%
```

```
  dplyr::filter(local_authority %in% unique(local_authority_characteristics$local_authority))

# Number of England local authorities with per 100k data
length(unique(england_hospital_visits_local_authority_per100k$local_authority))

# Number of England local authorities with per 100k data and complete matching data
length(unique(england_hospital_visits_local_authority_per100k_has_characteristics$local_authority))

if (mice_imputation == FALSE) {

# Filter local authority matching dataset
local_authority_characteristics_has_per100k = local_authority_characteristics %>%
  dplyr::filter(local_authority %in%
unique(england_hospital_visits_local_authority_per100k$local_authority) | local_authority %in%
unique(scotland_hospital_visits_local_authority_per100k$local_authority)) %>%
  dplyr::mutate(treatment = if_else(local_authority %in%
unique(scotland_hospital_visits_local_authority_per100k$local_authority),
                                    1, 0))

}

if (mice_imputation == TRUE) {

local_authority_characteristics_has_per100k_list = list()

for (i in 1:n_mice_imputations) {

  # Filter local authority matching dataset
  local_authority_characteristics_has_per100k = local_authority_characteristics_list[[i]] %>%
  dplyr::filter(local_authority %in%
unique(england_hospital_visits_local_authority_per100k$local_authority) | local_authority %in%
unique(scotland_hospital_visits_local_authority_per100k$local_authority)) %>%
  dplyr::mutate(treatment = if_else(local_authority %in%
unique(scotland_hospital_visits_local_authority_per100k$local_authority),
                                    1, 0))

  local_authority_characteristics_has_per100k_list[[i]] = local_authority_characteristics_has_per100k

}

}

# Number of Scotland and England local authorities with per 100k data and complete matching data
length(unique(local_authority_characteristics_has_per100k$local_authority))

# Create dataframe of Scotland and England outcomes
hospitalisations = scotland_hospital_visits_local_authority_per100k %>%
  dplyr::select(financial_year, year, local_authority, stays_per100k) %>%
  dplyr::filter(year >= 2012 & year <= 2019)

hospitalisations = rbind(hospitalisations,
england_hospital_visits_local_authority_per100k_has_characteristics)

```

```{r}

# Plot to check imputation was reasonable
s_plot = stripplot(mice_model, employment_rate_16to64, pch = 20, cex = 1.2, alpha = 0.75)

s_plot = update(s_plot,
                main = "Imputed Values (Red) for Employment Rate (16 to 64)",
                xlab = "Imputation Run (0 = Initial Dataset)",
                ylab = "Employment Rate")

s_plot

# Plot to check imputation was reasonable
s_plot = stripplot(mice_model, economic_activity_rate_16to64, pch = 20, cex = 1.2, alpha = 0.75)
```

````
s_plot = update(s_plot,
                main = "Imputed Values (Red) for Economic Activity Rate (16 to 64)",
                xlab = "Imputation Run (0 = Initial Dataset)",
                ylab = "Employment Rate")

s_plot

# Plot to check imputation was reasonable
s_plot = stripplot(mice_model, unemployment_rate_16to64, pch = 20, cex = 1.2, alpha = 0.75)

s_plot = update(s_plot,
                main = "Imputed Values (Red) for Unemployment Rate (16 to 64)",
                xlab = "Imputation Run (0 = Initial Dataset)",
                ylab = "Employment Rate")

s_plot

# Plot to check imputation was reasonable
s_plot = stripplot(mice_model, median_annual_pay_gross, pch = 20, cex = 1.2, alpha = 0.75)

s_plot = update(s_plot,
                main = "Imputed Values (Red) for Median Annual Gross Income",
                xlab = "Imputation Run (0 = Initial Dataset)",
                ylab = "Employment Rate")

s_plot

```

```{r}

# Scotland alcohol related hospital visits per 100,000 by local authority
ggplot(data = scotland_hospital_visits_local_authority_per100k, aes(x = year, y = stays_per100k,
colour = local_authority)) +
  geom_line(size = 0.75) +
  scale_x_continuous(breaks = breaks_width(4), minor_breaks = NULL) +
  scale_y_continuous(breaks = breaks_width(100)) +
  labs(x = "Year", y = "Hospital stays per 100,000", title = "Alcohol Related Hospital Visits per
100,000 by Local Authority - Scotland") +
  theme(panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.5), axis.text =
element_text(color = "black", size = 10),
        axis.title = element_text(size = 12), plot.title = element_text(size = 12, face = "bold"),
legend.title = element_blank())

```

```{r}

# Scotland alcohol related hospital visits per 100,000 by local authority
ggplot(data = scotland_hospital_visits_whole_per100k, aes(x = year, y = stays_per100k)) +
  geom_line(size = 0.75) +
  scale_x_continuous(breaks = breaks_width(4), minor_breaks = NULL) +
  scale_y_continuous(breaks = breaks_width(100), limits = c(0, 1000)) +
  labs(x = "Year", y = "Hospital stays per 100,000", title = "Alcohol Related Hospital Visits per
100,000 by Local Authority - Scotland") +
  theme(panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.5), axis.text =
element_text(color = "black", size = 10),
        axis.title = element_text(size = 12), plot.title = element_text(size = 12, face = "bold"),
legend.title = element_blank())

```

```{r}

# Visualise Scotland vs England total
scot_total = scotland_hospital_visits_whole_per100k %>%
  dplyr::rename(country = local_authority) %>%
  dplyr::select(c(country, year, stays_per100k)) %>%
````

````
  dplyr::filter(year >= 2012 & year <= 2019)

country = c("England", "England", "England", "England", "England", "England", "England", "England")
year = c(2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019)
stays_per100k = c(610, 620, 580, 560, 490, 490, 510, 520)

eng_total = data.frame(country, year, stays_per100k)

scot_eng_total = rbind(scot_total, eng_total)

# Scotland and England alcohol related hospital visits per 100,000 by local authority
ggplot(data = scot_eng_total, aes(x = year, y = stays_per100k, colour = country)) +
  geom_line(size = 0.75) +
  scale_x_continuous(breaks = breaks_width(1), minor_breaks = NULL) +
  scale_y_continuous(breaks = breaks_width(100), limits = c(0, 800)) +
  labs(x = "Year", y = "Hospital stays per 100,000", title = "Alcohol Related Hospital Visits per
100,000 Scotland vs England") +
  theme(panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.5), axis.text =
element_text(color = "black", size = 10),
        axis.title = element_text(size = 12), plot.title = element_text(size = 12, face = "bold"),
legend.title = element_blank())

```

```{r}

# Plot with pair correlations to justify Mahalanobis
covariates = local_authority_characteristics_has_per100k[, c("economic_activity_rate_16to64",
"employment_rate_16to64", "unemployment_rate_16to64",
                                                            "median_annual_pay_gross",
"percentage_male", "percentage_0to15",
                                                            "percentage_16to64",
"percentage_65plus")]

cor_matrix = cor(covariates, method = "spearman")

corrplot(cor_matrix, method = "color", type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 90, addCoef.col = "black", number.cex = 1)

```


```{r}

if (mice_imputation == FALSE) {

# Perform matching based on Mahalanobis distance
matchit_output = matchit(treatment ~ economic_activity_rate_16to64 + employment_rate_16to64 +
unemployment_rate_16to64 +
                           median_annual_pay_gross + percentage_male + percentage_0to15 +
percentage_16to64 +
                           percentage_65plus,
                         data = local_authority_characteristics_has_per100k,
                         method = "nearest", distance = "mahalanobis")

# Match data
matched_data = match.data(matchit_output)

# Match matrix
match_matrix = matchit_output$match.matrix

# Create matched pairs dataframe
matched_pairs = data.frame(Treatment_ID = rownames(match_matrix),
                           Control_ID = as.vector(match_matrix))

# Merge back into original data
matched_details = merge(local_authority_characteristics_has_per100k, matched_pairs, by.x =
"row.names", by.y = "Treatment_ID")
````

````
matched_details = merge(matched_details, local_authority_characteristics_has_per100k, by.x =
"Control_ID", by.y = "row.names", suffixes = c("_treat", "_control"))

# Just local authorities
matched_local_authorities = matched_details %>%
  dplyr::select(Control_ID, Row.names, local_authority_treat, local_authority_control)

}

```

```{r}

if (mice_imputation == TRUE) {

matched_local_authorities_list = list()

matched_data_list = list()

for (i in 1:n_mice_imputations) {
  # Perform matching based on Mahalanobis distance
  matchit_output = matchit(treatment ~ economic_activity_rate_16to64 + employment_rate_16to64 +
unemployment_rate_16to64 +
                             median_annual_pay_gross + percentage_male + percentage_0to15 +
percentage_16to64 +
                             percentage_65plus,
                           data = local_authority_characteristics_has_per100k_list[[i]],
                           method = "nearest", distance = "mahalanobis")

  # Match data
  matched_data = match.data(matchit_output)

  matched_data_list[[i]] = matched_data

  # Match matrix
  match_matrix = matchit_output$match.matrix

  # Create matched pairs dataframe
  matched_pairs = data.frame(Treatment_ID = rownames(match_matrix),
                             Control_ID = as.vector(match_matrix))

  # Merge back into original data
  matched_details = merge(local_authority_characteristics_has_per100k, matched_pairs, by.x =
"row.names", by.y = "Treatment_ID")
  matched_details = merge(matched_details, local_authority_characteristics_has_per100k, by.x =
"Control_ID", by.y = "row.names", suffixes = c("_treat", "_control"))

  # Just local authorities
  matched_local_authorities = matched_details %>%
    dplyr::select(Control_ID, Row.names, local_authority_treat, local_authority_control)

  matched_local_authorities_list[[i]] = matched_local_authorities

  }

}

```

```{r}

if (mice_imputation == TRUE) {

for (i in 1:n_mice_imputations) {

  matched_data_vis = matched_data_list[[i]] %>%
    dplyr::select(local_authority, treatment, economic_activity_rate_16to64, employment_rate_16to64,
unemployment_rate_16to64,
````

```
                  median_annual_pay_gross, percentage_male, percentage_0to15, percentage_16to64,
percentage_65plus) %>%
    dplyr::mutate(treatment = as.factor(treatment))

  # Histogram of distribution of Local AUthority variable treatment vs control
  plot = ggplot(data = matched_data_vis, aes(x = employment_rate_16to64, y = after_stat(count), fill =
treatment)) +
           geom_histogram(binwidth = 2.5, colour = "black", position = "identity", alpha = 0.3) +
           scale_fill_manual(values = c("1" = "red", "0" = "blue"), labels = c("1" = "Treatment
Group", "0" = "Control Group")) +
           scale_x_continuous(breaks = breaks_width(2.5), minor_breaks = NULL) +
           scale_y_continuous(breaks = breaks_width(2)) +
           labs(x = "Employment Rate Ages 16 to 64", y = "Count", title = "Distribution of Employment
Rate Ages 16 to 64 Treatment vs Control", fill = NULL) +
           theme(panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.5), axis.text
= element_text(color = "black", size = 10),
                 axis.title = element_text(size = 12), plot.title = element_text(size = 12, face =
"bold"))

  print(plot)

}

} else {

matched_data_vis = matched_data %>%
  dplyr::select(local_authority, treatment, economic_activity_rate_16to64, employment_rate_16to64,
unemployment_rate_16to64,
                median_annual_pay_gross, percentage_male, percentage_0to15, percentage_16to64,
percentage_65plus) %>%
  dplyr::mutate(treatment = as.factor(treatment))

# Histogram of distribution of Local AUthority variable treatment vs control
ggplot(data = matched_data_vis, aes(x = employment_rate_16to64, y = after_stat(count), fill =
treatment)) +
  geom_histogram(binwidth = 2.5, colour = "black", position = "identity", alpha = 0.3) +
  scale_fill_manual(values = c("1" = "red", "0" = "blue"), labels = c("1" = "Treatment Group", "0" =
"Control Group")) +
  scale_x_continuous(breaks = breaks_width(2.5), minor_breaks = NULL) +
  scale_y_continuous(breaks = breaks_width(2)) +
  labs(x = "Employment Rate Ages 16 to 64", y = "Count", title = "Distribution of Employment Rate Ages
16 to 64 Treatment vs Control", fill = NULL) +
  theme(panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.5), axis.text =
element_text(color = "black", size = 10),
        axis.title = element_text(size = 12), plot.title = element_text(size = 12, face = "bold"))

}
```

````
```{r}

if (mice_imputation == FALSE) {

# Basic pre vs post difference in difference
hospitalisations_wide = hospitalisations %>%
  dplyr::select(-financial_year) %>%
  tidyr::pivot_wider(names_from = year, values_from = stays_per100k)

dif_in_dif = matched_local_authorities %>%
  dplyr::left_join(hospitalisations_wide, by = c("local_authority_treat" = "local_authority")) %>%
  dplyr::left_join(hospitalisations_wide, by = c("local_authority_control" = "local_authority"),
suffix = c("_treat", "_control"))

# Pre and post treatment outcomes for the treatment and control groups
pre_treat = mean(dif_in_dif$`2017_treat`)
post_treat = mean(dif_in_dif$`2019_treat`)
pre_control = mean(dif_in_dif$`2017_control`)
post_control = mean(dif_in_dif$`2019_control`)
````
````

````
dif_in_dif_estimate = (post_treat - pre_treat) - (post_control - pre_control)

dif_in_dif_estimate

}

```

```{r}

if (mice_imputation == TRUE) {

dif_in_dif_estimate_list = c()

dif_in_dif_list = list()

for (i in 1:n_mice_imputations) {

  # Basic pre vs post difference in difference
  hospitalisations_wide = hospitalisations %>%
    dplyr::select(-financial_year) %>%
    tidyr::pivot_wider(names_from = year, values_from = stays_per100k)

  dif_in_dif = matched_local_authorities_list[[i]] %>%
    dplyr::left_join(hospitalisations_wide, by = c("local_authority_treat" = "local_authority")) %>%
    dplyr::left_join(hospitalisations_wide, by = c("local_authority_control" = "local_authority"),
suffix = c("_treat", "_control"))

  # Pre and post treatment outcomes for the treatment and control groups
  pre_treat = mean(dif_in_dif$`2017_treat`)
  post_treat = mean(dif_in_dif$`2019_treat`)
  pre_control = mean(dif_in_dif$`2017_control`)
  post_control = mean(dif_in_dif$`2019_control`)

  dif_in_dif_estimate = (post_treat - pre_treat) - (post_control - pre_control)

  dif_in_dif_estimate

  dif_in_dif_estimate_list[[i]] = dif_in_dif_estimate

  dif_in_dif_list[[i]] = dif_in_dif

}

  dif_in_dif_estimate_list

}

```

```{r}

if (mice_imputation == FALSE) {

# Test parallel trends assumption
hospitalisations_long = dif_in_dif %>%
  tidyr::pivot_longer(cols = starts_with("20"),
                      names_to = "year",
                      values_to = "stays_per100k") %>%
  dplyr::mutate(group = case_when(str_ends(year, "_treat") ~ "Treatment",
                                  str_ends(year, "_control") ~ "Control",
                                  TRUE ~ NA)) %>%
  dplyr::mutate(year = as.numeric(str_sub(str_trim(year), start = 1, end = 4)) + 1) %>%
  dplyr::group_by(group, year) %>%
  dplyr::summarise(stays_per100k = mean(stays_per100k))

# Treatment and control alcohol related hospital visits per 100,000
ggplot(data = hospitalisations_long, aes(x = year, y = stays_per100k, colour = group)) +
````

````
  geom_line(size = 0.75) +
  geom_vline(xintercept = 2018.33, color = "red", size = 0.6, alpha = 0.8) +
  annotate("text", x = 2019.05, y = 25, label = "Intervention", color = "red", size = 4) +
  geom_vline(xintercept = 2017.92, color = "blue", size = 0.6, alpha = 0.8) +
  annotate("text", x = 2016.425, y = 25, label = "Intervention Announcement", color = "blue", size =
4) +
  scale_x_continuous(breaks = breaks_width(1), minor_breaks = NULL) +
  scale_y_continuous(breaks = breaks_width(100), limits = c(0, 800)) +
  labs(x = "Year", y = "Mean Hospital stays per 100,000", title = "Alcohol Related Hospital Visits per
100,000 Treatment vs Control") +
  theme(panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.5), axis.text =
element_text(color = "black", size = 10),
        axis.title = element_text(size = 12), plot.title = element_text(size = 12, face = "bold"),
legend.title = element_blank())

}

```

```{r}

if (mice_imputation == TRUE) {

for (i in 1:n_mice_imputations) {

  # Test parallel trends assumption
  hospitalisations_long = dif_in_dif_list[[i]] %>%
    tidyr::pivot_longer(cols = starts_with("20"),
                        names_to = "year",
                        values_to = "stays_per100k") %>%
    dplyr::mutate(group = case_when(str_ends(year, "_treat") ~ "Treatment",
                                    str_ends(year, "_control") ~ "Control",
                                    TRUE ~ NA)) %>%
    dplyr::mutate(year = as.numeric(str_sub(str_trim(year), start = 1, end = 4)) + 1) %>%
    dplyr::group_by(group, year) %>%
    dplyr::summarise(stays_per100k = mean(stays_per100k))

  # Treatment and control alcohol related hospital visits per 100,000
  plot = ggplot(data = hospitalisations_long, aes(x = year, y = stays_per100k, colour = group)) +
           geom_line(size = 0.75) +
           geom_vline(xintercept = 2018.33, color = "black", size = 0.6, alpha = 0.8) +
           annotate("text", x = 2019.05, y = 25, label = "Intervention", color = "black", size = 4) +
           geom_vline(xintercept = 2017.92, color = "#333333", size = 0.6, alpha = 0.8, linetype =
"dashed") +
           annotate("text", x = 2016.425, y = 25, label = "Intervention Announcement", color =
"#333333", size = 4) +
           scale_x_continuous(breaks = breaks_width(1), minor_breaks = NULL) +
           scale_y_continuous(breaks = breaks_width(100), limits = c(0, 800)) +
           scale_color_manual(values = c("Treatment" = "#EC4646", "Control" = "#4646EC")) +
           labs(x = "Year", y = "Mean Hospital stays per 100,000", title = "Alcohol Related Hospital
Visits per 100,000 Treatment vs Control") +
           theme(panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.5), axis.text
= element_text(color = "black", size = 10),
                 axis.title = element_text(size = 12), plot.title = element_text(size = 12, face =
"bold"), legend.title = element_blank())

  print(plot)

}

}

```

```{r}

if (mice_imputation == TRUE) {

for (i in 1:n_mice_imputations) {
````

````
  if (i == 1) {

    # Test parallel trends assumption
    hospitalisations_long = dif_in_dif_list[[i]] %>%
      tidyr::pivot_longer(cols = starts_with("20"),
                          names_to = "year",
                          values_to = "stays_per100k") %>%
      dplyr::mutate(group = case_when(str_ends(year, "_treat") ~ "Treatment",
                                      str_ends(year, "_control") ~ "Control",
                                      TRUE ~ NA)) %>%
      dplyr::mutate(year = as.numeric(str_sub(str_trim(year), start = 1, end = 4)) + 1) %>%
      dplyr::group_by(group, year) %>%
      dplyr::summarise(stays_per100k = mean(stays_per100k), .groups = "drop") %>%
      dplyr::mutate(facet = i)

  }

  # Test parallel trends assumption
  hospitalisations_next = dif_in_dif_list[[i]] %>%
    tidyr::pivot_longer(cols = starts_with("20"),
                        names_to = "year",
                        values_to = "stays_per100k") %>%
    dplyr::mutate(group = case_when(str_ends(year, "_treat") ~ "Treatment",
                                    str_ends(year, "_control") ~ "Control",
                                    TRUE ~ NA)) %>%
    dplyr::mutate(year = as.numeric(str_sub(str_trim(year), start = 1, end = 4)) + 1) %>%
    dplyr::group_by(group, year) %>%
    dplyr::summarise(stays_per100k = mean(stays_per100k), .groups = "drop") %>%
    dplyr::mutate(facet = i)

  hospitalisations_long = rbind(hospitalisations_long, hospitalisations_next)

}

hospitalisations_long$group = factor(hospitalisations_long$group, levels = c("Treatment", "Control",
"Intervention Announcement", "Intervention"))

legend_order = factor(c("Treatment", "Control", "Intervention Announcement", "Intervention"), levels =
c("Intervention", "Intervention Announcement", "Treatment", "Control"))

# Plot parallel trends all on one graph
ggplot(data = hospitalisations_long, aes(x = year, y = stays_per100k, colour = group)) +
  geom_line(size = 0.75) +
  geom_vline(aes(xintercept = 2018.33, color = "Intervention"), size = 0.6, alpha = 0.8) +
  geom_vline(aes(xintercept = 2017.92, color = "Intervention Announcement"), size = 0.6, alpha = 0.8,
linetype = "dashed") +
  scale_color_manual(values = c("Intervention" = "red", "Intervention Announcement" = "blue",
"Treatment" = "#00BFC4", "Control" = "#F8766D")) +
  scale_x_continuous(breaks = breaks_width(1), minor_breaks = NULL) +
  scale_y_continuous(breaks = breaks_width(100), limits = c(0, 800)) +
  scale_color_manual(values = c("Treatment" = "#EC4646", "Control" = "#4646EC", "Intervention
Announcement" = "#333333", "Intervention" = "black")) +
  labs(x = "Year", y = "Mean Hospital stays per 100,000", title = "Alcohol Related Hospital Visits per
100,000 Treatment vs Control",
       color = "Legend") +
  theme(panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.5),
        axis.text = element_text(color = "black", size = 8),
        axis.text.x = element_text(angle = 90, hjust = 1),
        axis.title = element_text(size = 12),
        plot.title = element_text(size = 12, face = "bold"),
        legend.title = element_blank()) +
  facet_wrap(~facet)

}

```

```{r}
````

```
if (mice_imputation == FALSE) {

# Placebo test for every two year time period prior and spanning the intervention date
dif_in_dif_2y_estimates = list()

# Dif in dif calculation for each placebo and actual estimate
for (i in 1:6) {

  pre_treat_i = mean(dif_in_dif[, (i + 4)])
  post_treat_i = mean(dif_in_dif[, (i + 6)])
  pre_control_i = mean(dif_in_dif[, (i + 12)])
  post_control_i = mean(dif_in_dif[, (i + 14)])

  dif_in_dif_estimate_i = (post_treat_i - pre_treat_i) - (post_control_i - pre_control_i)

  dif_in_dif_2y_estimates[[i]] = dif_in_dif_estimate_i

}

# Function to compute dif in dif estimate
dif_in_dif_function = function(data, indices) {

  # Bootstrap resampling
  data_subset = data[indices, ]

  # Dif in dif calculation for each placebo and actual estimate
  estimates = sapply(1:6, function(i) {

    pre_treat_i = mean(data_subset[, (i + 4)])
    post_treat_i = mean(data_subset[, (i + 6)])
    pre_control_i = mean(data_subset[, (i + 12)])
    post_control_i = mean(data_subset[, (i + 14)])

    (post_treat_i - pre_treat_i) - (post_control_i - pre_control_i)

  })

  return(estimates)

}

# Bootstrap to calculate standard errors for causal estimate and placebo causal estimates
set.seed(12345)

bootstrap_results = boot(data = dif_in_dif, statistic = dif_in_dif_function, R = 10000)

# Standard errors
boot_se = apply(bootstrap_results$t, 2, sd)

# Bootstrap confidence intervals
boot_ci = t(apply(bootstrap_results$t, 2, function(x) {

  quantile(x, probs = c(0.025, 0.975))

}))

# Dataframe rows
dif_in_dif_2y_estimates = unlist(dif_in_dif_2y_estimates)
year = c(2015.25, 2016.25, 2017.25, 2018.25, 2019.25, 2020.25) # End dates of years for showing
treatment date against
lower_ci = boot_ci[, 1]
upper_ci = boot_ci[, 2]
se = boot_se

# Create dataframe
dif_in_dif_2y_estimates_placebo = data.frame(year, dif_in_dif_2y_estimates, lower_ci, upper_ci, se)

# Calculate p-values
```

````
dif_in_dif_2y_estimates_placebo = dif_in_dif_2y_estimates_placebo %>%
  dplyr::mutate(p_value = 2 * pnorm(-abs(dif_in_dif_2y_estimates / se)))

# Placebo test results for every two year time period prior to and spanning the intervention date
ggplot(data = dif_in_dif_2y_estimates_placebo, aes(x = year, y = dif_in_dif_2y_estimates)) +
  geom_line(size = 1, linetype = "dotted", color = "black", size = 0.75, alpha = 0.75) +
  geom_point(shape = 16, size = 3, color = "black") +
  geom_errorbar(aes(ymin = lower_ci, ymax = upper_ci), width = 0.1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "dark grey", size = 0.75, alpha = 0.8) +
  geom_vline(xintercept = 2018.33, color = "red", size = 0.6, alpha = 0.8) +
  annotate("text", x = 2018.75, y = -75.5, label = "Intervention", color = "red", size = 4) +
  geom_vline(xintercept = 2017.92, color = "blue", size = 0.6, alpha = 0.8) +
  annotate("text", x = 2017.05, y = -75.5, label = "Intervention Announcement", color = "blue", size =
4) +
  scale_x_continuous(breaks = breaks_width(1), minor_breaks = NULL) +
  scale_y_continuous(breaks = breaks_width(10), limits = c(-80, 80)) +
  labs(x = "Year", y = "Dif in Dif Estimated Placebo Treatment Effects", title = "Estimated Causal
Effect Including Pretreatment Placebos") +
  theme(panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.5), axis.text =
element_text(color = "black", size = 10),
        axis.title = element_text(size = 12), plot.title = element_text(size = 12, face = "bold"),
legend.title = element_blank())

}

```

```{r}

if (mice_imputation == FALSE) {

# Placebo test for every one year time period prior and spanning the intervention date
dif_in_dif_1y_estimates = list()

# Dif in dif calculation for each placebo and actual estimate
for (i in 1:6) {

  pre_treat_i = mean(dif_in_dif[, (i + 4)])
  post_treat_i = mean(dif_in_dif[, (i + 5)])
  pre_control_i = mean(dif_in_dif[, (i + 12)])
  post_control_i = mean(dif_in_dif[, (i + 13)])

  dif_in_dif_estimate_i = (post_treat_i - pre_treat_i) - (post_control_i - pre_control_i)

  dif_in_dif_1y_estimates[[i]] = dif_in_dif_estimate_i

}

# Function to compute dif in dif estimate
dif_in_dif_function = function(data, indices) {

  # Bootstrap resampling
  data_subset = data[indices, ]

  # Dif in dif calculation for each placebo and actual estimate
  estimates = sapply(1:6, function(i) {

    pre_treat_i = mean(data_subset[, (i + 4)])
    post_treat_i = mean(data_subset[, (i + 5)])
    pre_control_i = mean(data_subset[, (i + 12)])
    post_control_i = mean(data_subset[, (i + 13)])

    (post_treat_i - pre_treat_i) - (post_control_i - pre_control_i)

  })

  return(estimates)

}
````

```
# Bootstrap to calculate standard errors for causal estimate and placebo causal estimates
set.seed(12345)

bootstrap_results = boot(data = dif_in_dif, statistic = dif_in_dif_function, R = 10000)

# Standard errors
boot_se = apply(bootstrap_results$t, 2, sd)

# Bootstrap confidence intervals
boot_ci = t(apply(bootstrap_results$t, 2, function(x) {

  quantile(x, probs = c(0.025, 0.975))

}))

# Dataframe rows
dif_in_dif_1y_estimates = unlist(dif_in_dif_1y_estimates)
year = c(2014.25, 2015.25, 2016.25, 2017.25, 2018.25, 2019.25)
lower_ci = boot_ci[, 1]
upper_ci = boot_ci[, 2]
se = boot_se

# Create dataframe
dif_in_dif_1y_estimates_placebo = data.frame(year, dif_in_dif_1y_estimates, lower_ci, upper_ci, se)

# Calculate p-values
dif_in_dif_1y_estimates_placebo = dif_in_dif_1y_estimates_placebo %>%
  dplyr::mutate(p_value = 2 * pnorm(-abs(dif_in_dif_1y_estimates / se)))

# Placebo test results for every one year time period prior to and spanning the intervention date
ggplot(data = dif_in_dif_1y_estimates_placebo, aes(x = year, y = dif_in_dif_1y_estimates)) +
  geom_line(size = 1, linetype = "dotted", color = "black", size = 0.75, alpha = 0.75) +
  geom_point(shape = 16, size = 3, color = "black") +
  geom_errorbar(aes(ymin = lower_ci, ymax = upper_ci), width = 0.1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "dark grey", size = 0.75, alpha = 0.8) +
  geom_vline(xintercept = 2018.33, color = "red", size = 0.6, alpha = 0.8) +
  annotate("text", x = 2018.75, y = -57, label = "Intervention", color = "red", size = 4) +
  geom_vline(xintercept = 2017.92, color = "blue", size = 0.6, alpha = 0.8) +
  annotate("text", x = 2017.05, y = -57, label = "Intervention Announcement", color = "blue", size =
4) +
  scale_x_continuous(breaks = breaks_width(1), minor_breaks = NULL) +
  scale_y_continuous(breaks = breaks_width(10), limits = c(-60, 60)) +
  labs(x = "Year", y = "Dif in Dif Estimated Placebo Treatment Effects", title = "Alcohol Related
Hospital Visits per 100,000 Treatment vs Control") +
  theme(panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.5), axis.text =
element_text(color = "black", size = 10),
        axis.title = element_text(size = 12), plot.title = element_text(size = 12, face = "bold"),
legend.title = element_blank())

}
```

```{r}

if (mice_imputation == TRUE) {

dif_in_dif_2y_estimates_placebo_list = list()

for (i in 1:n_mice_imputations) {

  # Placebo test for every two year time period prior and spanning the intervention date
  dif_in_dif_2y_estimates = list()

  # Dif in dif calculation for each placebo and actual estimate
  for (j in 1:6) {

    pre_treat_j = mean(dif_in_dif_list[[i]][, (j + 4)])
```

```
    post_treat_j = mean(dif_in_dif_list[[i]][, (j + 6)])
    pre_control_j = mean(dif_in_dif_list[[i]][, (j + 12)])
    post_control_j = mean(dif_in_dif_list[[i]][, (j + 14)])

    dif_in_dif_estimate_j = (post_treat_j - pre_treat_j) - (post_control_j - pre_control_j)

    dif_in_dif_2y_estimates[[j]] = dif_in_dif_estimate_j

  }

  # Function to compute dif in dif estimate
  dif_in_dif_function = function(data, indices) {

    # Bootstrap resampling
    data_subset = data[indices, ]

    # Dif in dif calculation for each placebo and actual estimate
    estimates = sapply(1:6, function(k) {

      pre_treat_k = mean(data_subset[, (k + 4)])
      post_treat_k = mean(data_subset[, (k + 6)])
      pre_control_k = mean(data_subset[, (k + 12)])
      post_control_k = mean(data_subset[, (k + 14)])

      (post_treat_k - pre_treat_k) - (post_control_k - pre_control_k)

    })

    return(estimates)

  }

  # Bootstrap to calculate standard errors for causal estimate and placebo causal estimates
  set.seed(12345)

  bootstrap_results = boot(data = dif_in_dif_list[[i]], statistic = dif_in_dif_function, R = 10000)

  # Standard errors
  boot_se = apply(bootstrap_results$t, 2, sd)

  # Bootstrap confidence intervals
  boot_ci = t(apply(bootstrap_results$t, 2, function(x) {

    quantile(x, probs = c(0.025, 0.975))

  }))

  # Dataframe rows
  dif_in_dif_2y_estimates = unlist(dif_in_dif_2y_estimates)
  year = c(2015.25, 2016.25, 2017.25, 2018.25, 2019.25, 2020.25) # End dates of years for showing
treatment date against
  lower_ci = boot_ci[, 1]
  upper_ci = boot_ci[, 2]
  se = boot_se

  # Create dataframe
  dif_in_dif_2y_estimates_placebo = data.frame(year, dif_in_dif_2y_estimates, lower_ci, upper_ci, se)

  # Calculate p-values
  dif_in_dif_2y_estimates_placebo = dif_in_dif_2y_estimates_placebo %>%
    dplyr::mutate(p_value = 2 * pnorm(-abs(dif_in_dif_2y_estimates / se)))

  dif_in_dif_2y_estimates_placebo_list[[i]] = dif_in_dif_2y_estimates_placebo

}

for (i in 1:n_mice_imputations) {

  if (i == 1) {
```

````
    dif_in_dif_2y_estimates_placebo_df = dif_in_dif_2y_estimates_placebo_list[[i]] %>%
      dplyr::mutate(mice_run = i) %>%
      dplyr::mutate(mice_run = as.character(mice_run))

  }

  if (i != 1) {

    dif_in_dif_2y_estimates_placebo_i = dif_in_dif_2y_estimates_placebo_list[[i]] %>%
      dplyr::mutate(mice_run = i) %>%
      dplyr::mutate(mice_run = as.character(mice_run))

    dif_in_dif_2y_estimates_placebo_df = rbind(dif_in_dif_2y_estimates_placebo_df,
dif_in_dif_2y_estimates_placebo_i)

  }

}

# Rubin's Rules to get overall standard error, confidence interval and p-value estimates
dif_in_dif_2y_estimates_overall = dif_in_dif_2y_estimates_placebo_df %>%
  dplyr::group_by(year) %>%
  dplyr::summarise(mean_estimate = mean(dif_in_dif_2y_estimates),
                   within_imputation_variance = mean((se)^2),
                   between_imputation_variance = var(dif_in_dif_2y_estimates),
                   total_variance = mean((se)^2) + (1 + (1 / n_mice_imputations)) *
var(dif_in_dif_2y_estimates),
                   se_pooled = sqrt(mean((se)^2) + (1 + (1 / n_mice_imputations)) *
var(dif_in_dif_2y_estimates)),
                   lower_ci = mean(dif_in_dif_2y_estimates) - 1.96 * sqrt(mean((se)^2) + (1 + (1 /
n_mice_imputations)) * var(dif_in_dif_2y_estimates)),
                   upper_ci = mean(dif_in_dif_2y_estimates) + 1.96 * sqrt(mean((se)^2) + (1 + (1 /
n_mice_imputations)) * var(dif_in_dif_2y_estimates)),
                   p_value = 2 * (1 - pnorm(abs(mean(dif_in_dif_2y_estimates) / sqrt(mean((se)^2) + (1
+ (1 / n_mice_imputations)) * var(dif_in_dif_2y_estimates))))))

# Placebo test results for every two year time period prior to and spanning the intervention date
ggplot(data = dif_in_dif_2y_estimates_overall, aes(x = year, y = mean_estimate)) +
  geom_line(size = 1, linetype = "dotted", color = "black", size = 0.75, alpha = 0.75) +
  geom_point(shape = 16, size = 3, color = "black") +
  geom_errorbar(aes(ymin = lower_ci, ymax = upper_ci), width = 0.1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "dark grey", size = 0.75, alpha = 0.8) +
  geom_vline(xintercept = 2018.33, color = "red", size = 0.6, alpha = 0.8) +
  annotate("text", x = 2018.75, y = -75.5, label = "Intervention", color = "red", size = 4) +
  geom_vline(xintercept = 2017.92, color = "#333333", size = 0.6, alpha = 0.8, linetype = "dashed") +
  annotate("text", x = 2017.05, y = -75.5, label = "Intervention Announcement", color = "#333333",
size = 4) +
  scale_x_continuous(breaks = breaks_width(1), minor_breaks = NULL) +
  scale_y_continuous(breaks = breaks_width(10), limits = c(-80, 80)) +
  labs(x = "Year", y = "Dif in Dif Estimated Placebo Treatment Effects", title = "Estimated Causal
Effect 2-Year Pretreatment Placebos") +
  theme(panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.5), axis.text =
element_text(color = "black", size = 10),
        axis.title = element_text(size = 12), plot.title = element_text(size = 12, face = "bold"),
legend.title = element_blank())

}

```

```{r}

if (mice_imputation == TRUE) {

dif_in_dif_1y_estimates_placebo_list = list()

for (i in 1:n_mice_imputations) {
````

```
# Placebo test for every one year time period prior and spanning the intervention date
dif_in_dif_1y_estimates = list()

# Dif in dif calculation for each placebo and actual estimate
for (j in 1:6) {

  pre_treat_j = mean(dif_in_dif_list[[i]][, (j + 4)])
  post_treat_j = mean(dif_in_dif_list[[i]][, (j + 5)])
  pre_control_j = mean(dif_in_dif_list[[i]][, (j + 12)])
  post_control_j = mean(dif_in_dif_list[[i]][, (j + 13)])

  dif_in_dif_estimate_j = (post_treat_j - pre_treat_j) - (post_control_j - pre_control_j)

  dif_in_dif_1y_estimates[[j]] = dif_in_dif_estimate_j

}

# Function to compute dif in dif estimate
dif_in_dif_function = function(data, indices) {

  # Bootstrap resampling
  data_subset = data[indices, ]

  # Dif in dif calculation for each placebo and actual estimate
  estimates = sapply(1:6, function(k) {

    pre_treat_k = mean(data_subset[, (k + 4)])
    post_treat_k = mean(data_subset[, (k + 6)])
    pre_control_k = mean(data_subset[, (k + 12)])
    post_control_k = mean(data_subset[, (k + 14)])

    (post_treat_k - pre_treat_k) - (post_control_k - pre_control_k)

  })

  return(estimates)

}

# Bootstrap to calculate standard errors for causal estimate and placebo causal estimates
set.seed(12345)

bootstrap_results = boot(data = dif_in_dif_list[[i]], statistic = dif_in_dif_function, R = 10000)

# Standard errors
boot_se = apply(bootstrap_results$t, 2, sd)

# Bootstrap confidence intervals
boot_ci = t(apply(bootstrap_results$t, 2, function(x) {

  quantile(x, probs = c(0.025, 0.975))

}))

# Dataframe rows
dif_in_dif_1y_estimates = unlist(dif_in_dif_1y_estimates)
year = c(2014.25, 2015.25, 2016.25, 2017.25, 2018.25, 2019.25) # End dates of years for showing
treatment date against
lower_ci = boot_ci[, 1]
upper_ci = boot_ci[, 2]
se = boot_se

# Create dataframe
dif_in_dif_1y_estimates_placebo = data.frame(year, dif_in_dif_1y_estimates, lower_ci, upper_ci, se)

# Calculate p-values
dif_in_dif_1y_estimates_placebo = dif_in_dif_1y_estimates_placebo %>%
  dplyr::mutate(p_value = 2 * pnorm(-abs(dif_in_dif_1y_estimates / se)))
```

```
  dif_in_dif_1y_estimates_placebo_list[[i]] = dif_in_dif_1y_estimates_placebo

}

for (i in 1:n_mice_imputations) {

  if (i == 1) {

    dif_in_dif_1y_estimates_placebo_df = dif_in_dif_1y_estimates_placebo_list[[i]] %>%
      dplyr::mutate(mice_run = i) %>%
      dplyr::mutate(mice_run = as.character(mice_run))

  }

  if (i != 1) {

    dif_in_dif_1y_estimates_placebo_i = dif_in_dif_1y_estimates_placebo_list[[i]] %>%
      dplyr::mutate(mice_run = i) %>%
      dplyr::mutate(mice_run = as.character(mice_run))

    dif_in_dif_1y_estimates_placebo_df = rbind(dif_in_dif_1y_estimates_placebo_df,
dif_in_dif_1y_estimates_placebo_i)

  }

}

# Rubin's Rules to get overall standard error, confidence interval and p-value estimates
dif_in_dif_1y_estimates_overall = dif_in_dif_1y_estimates_placebo_df %>%
  dplyr::group_by(year) %>%
  dplyr::summarise(mean_estimate = mean(dif_in_dif_1y_estimates),
                   within_imputation_variance = mean((se)^2),
                   between_imputation_variance = var(dif_in_dif_1y_estimates),
                   total_variance = mean((se)^2) + (1 + (1 / n_mice_imputations)) *
var(dif_in_dif_1y_estimates),
                   se_pooled = sqrt(mean((se)^2) + (1 + (1 / n_mice_imputations)) *
var(dif_in_dif_1y_estimates)),
                   lower_ci = mean(dif_in_dif_1y_estimates) - 1.96 * sqrt(mean((se)^2) + (1 + (1 /
n_mice_imputations)) * var(dif_in_dif_1y_estimates)),
                   upper_ci = mean(dif_in_dif_1y_estimates) + 1.96 * sqrt(mean((se)^2) + (1 + (1 /
n_mice_imputations)) * var(dif_in_dif_1y_estimates)),
                   p_value = 2 * (1 - pnorm(abs(mean(dif_in_dif_1y_estimates) / sqrt(mean((se)^2) + (1
+ (1 / n_mice_imputations)) * var(dif_in_dif_1y_estimates))))))

# Placebo test results for every one year time period prior to and spanning the intervention date
ggplot(data = dif_in_dif_1y_estimates_overall, aes(x = year, y = mean_estimate)) +
  geom_line(size = 1, linetype = "dotted", color = "black", size = 0.75, alpha = 0.75) +
  geom_point(shape = 16, size = 3, color = "black") +
  geom_errorbar(aes(ymin = lower_ci, ymax = upper_ci), width = 0.1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "dark grey", size = 0.75, alpha = 0.8) +
  geom_vline(xintercept = 2018.33, color = "red", size = 0.6, alpha = 0.8) +
  annotate("text", x = 2018.75, y = -75.5, label = "Intervention", color = "red", size = 4) +
  geom_vline(xintercept = 2017.92, color = "#333333", size = 0.6, alpha = 0.8, linetype = "dashed") +
  annotate("text", x = 2017.05, y = -75.5, label = "Intervention Announcement", color = "#333333",
size = 4) +
  scale_x_continuous(breaks = breaks_width(1), minor_breaks = NULL) +
  scale_y_continuous(breaks = breaks_width(10), limits = c(-80, 80)) +
  labs(x = "Year", y = "Dif in Dif Estimated Placebo Treatment Effects", title = "Estimated Causal
Effect 1-Year Pretreatment Placebos") +
  theme(panel.border = element_rect(colour = "black", fill = NA, linewidth = 0.5), axis.text =
element_text(color = "black", size = 10),
        axis.title = element_text(size = 12), plot.title = element_text(size = 12, face = "bold"),
legend.title = element_blank())

}
```