**Appendix for causal inference paper with explanations and R codes**

**Chan et al (2022)**

**This document contains:**

* Appendix 1. Matching example in R.
* Appendix 2. Inverse probability treatment weighting (cross-sectional/ pre-post design)
* Appendix 3. Inverse probability treatment weighting (multi-wave longitudinal design)
* Appendix 4. Interrupted time series analysis with control

|  |
| --- |
| All codes in this document have been tested on R 4.2.1, and it might not be compatible with older version of R. If errors are encountered when running any codes, try downloading the latest version of R from <https://r-project.org> and reinstall all the required packages using the “install.packages()” function (see below).  Different versions of R packages may produce slightly different results.  The explanations are colour-coded in black.  R codes are presented in purple.  R results and output are presented in blue.  The R codes for all the examples can be downloaded from <https://github.com/gckc123/Causal_Analysis_Addiction_Examples> |

# Appendix 1. Matching example in R.

In this example, we will use a dataset simulated based on a national survey in Australia to test the association between smoking and psychological distress.

**Installing and loading necessary R libraries**

The libraries “tidyverse”, “lmtest”, “sandwich” and “MatchIt” will be used in this example. If they are not yet installed, they can be installed using the codes below.

install.packages("tidyverse", dependencies = TRUE)  
install.packages("lmtest", dependencies = TRUE)  
install.packages("sandwich", dependencies = TRUE)  
install.packages("MatchIt", dependencies = TRUE)

After installing these libraries, the following codes can be used to load these libraries.

library("tidyverse")  
library("lmtest")  
library("sandwich")  
library("MatchIt")

**Loading the dataset**

The data is available at Github and can be loaded using the following codes.

smk\_data <- read\_csv("https://raw.githubusercontent.com/gckc123/Causal\_Analysis\_Addiction\_Examples/main/smoking\_psyc\_distress.csv")

This dataset contains 10 variables  
  
**sex:** 0: Female; 1: Male  
**indigeneity:** 0: non-indigenous; 1: indigenous  
**high\_school:** 0: not completed high school; 1: completed high school  
**partnered:** 0: not partnered; 1: partnered  
**remoteness:** Remoteness of an individual’s residence. 0: major cities; 1: inner regional; 2: outer regional or more remote area  
**language:** Main language of the participant. 0: non-English; 1: English  
**smoker:** 0: No; 1: Yes  
**risky\_alcohol:** Consuming alcohol at a risky level. 0: No; 1: Yes  
**psyc\_distress:** Numeric variable ranged from 10 to 50. Higher value represents higher level of psychological distress  
**age:** Age of the participant

Since remoteness is a categorical variable with 3 levels, we need to convert it into a ‘factor’ variable so that R knows that it is a categorical variable.

smk\_data$remoteness <- as.factor(smk\_data$remoteness)

**Matching and evaluating balance between treatment and control group**

Suppose we are testing the association between the smoker and psychological distress variables, adjusting for all other social demographic variables in the dataset, and we are interested in estimating the *average treatment effect among the treated* (ATT; because smoking can be considered as a self-selected “treatment”). The following codes can be used. We have specified using optimal matching by specifying “optimal” as the method parameter, and using generalized linear model to estimate the propensity score for matching by specifying “glm” as the distance parameter.   
  
The performance of these settings is often sufficient. Other options for matching include “nearest” (greedy matching) and “full” (full optimal matching); other options for distance includes “gam” (generalised additive model) and “gbm” (generalised boosted model). Interested readers can use the code “?matchit” to invoke the help page for the matchit function for more detailed explanations.

smk\_matching <- matchit(smoker ~ sex + indigeneity + high\_school + partnered + remoteness + language + risky\_alcohol + age, data = smk\_data, method = "optimal", distance = "glm")

summary(smk\_matching)  
  
The above codes produce the output below.

Call:  
matchit(formula = smoker ~ sex + indigeneity + high\_school +   
 partnered + remoteness + language + risky\_alcohol + age,   
 data = smk\_data, method = "optimal", distance = "glm")  
  
Summary of Balance for All Data:  
 Means Treated Means Control **Std. Mean Diff.** Var. Ratio eCDF Mean eCDF Max  
distance 0.1852 0.1130 **0.6304** 2.0778 0.2033 0.3185  
sex 0.4938 0.4421 **0.1035** . 0.0518 0.0518  
indigeneity 0.0524 0.0175 **0.1565** . 0.0349 0.0349  
high\_school 0.4220 0.6378 **-0.4370** . 0.2158 0.2158  
partnered 0.4630 0.6913 **-0.4578** . 0.2283 0.2283  
remoteness0 0.5852 0.6773 **-0.1870** . 0.0921 0.0921  
remoteness1 0.2177 0.1900 **0.0670** . 0.0277 0.0277  
remoteness2 0.1971 0.1327 **0.1621** . 0.0645 0.0645  
language 0.9579 0.9130 **0.2234** . 0.0449 0.0449  
risky\_alcohol 0.6427 0.5411 **0.2120** . 0.1016 0.1016  
age 51.6057 53.7824 **-0.1676** 0.8214 0.0441 0.1020

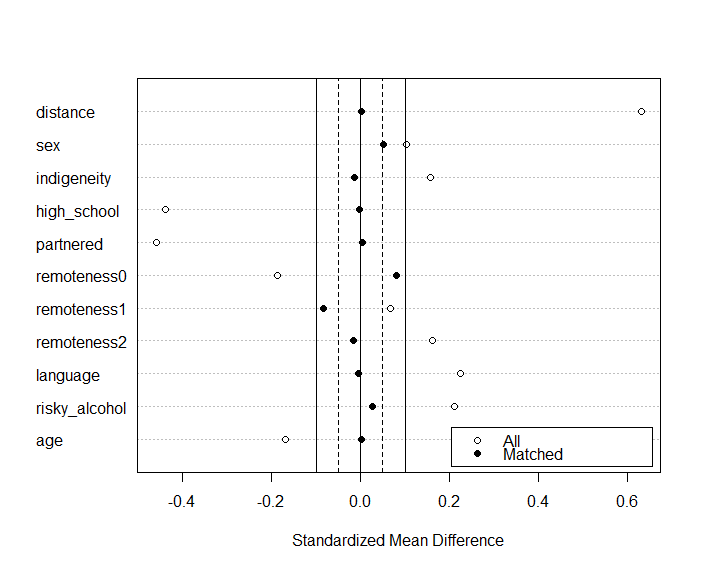
Summary of Balance for Matched Data:  
 Means Treated Means Control **Std. Mean Diff.** Var. Ratio eCDF Mean eCDF Max Std. Pair Dist.  
distance 0.1852 0.1849 **0.0024** 1.0154 0.0010 0.0072 0.0077  
sex 0.4938 0.4682 **0.0513** . 0.0257 0.0257 0.9467  
indigeneity 0.0524 0.0554 **-0.0138** . 0.0031 0.0031 0.3918  
high\_school 0.4220 0.4230 **-0.0021** . 0.0010 0.0010 0.6341  
partnered 0.4630 0.4610 **0.0041** . 0.0021 0.0021 0.6548  
remoteness0 0.5852 0.5452 **0.0813** . 0.0400 0.0400 0.9357  
remoteness1 0.2177 0.2515 **-0.0821** . 0.0339 0.0339 0.8882  
remoteness2 0.1971 0.2033 **-0.0155** . 0.0062 0.0062 0.7742  
language 0.9579 0.9589 **-0.0051** . 0.0010 0.0010 0.3630  
risky\_alcohol 0.6427 0.6294 **0.0279** . 0.0133 0.0133 0.9106  
age 51.6057 51.5729 **0.0025** 0.8406 0.0242 0.0462 1.0836

Sample Sizes:  
 Control Treated  
All 7026 974  
Matched **974 974**  
Unmatched 6052 0  
Discarded 0 0

The first table shows the difference in each of the variables between smokers and non-smokers. All standardised mean differences are over 0.1, indicating that there are substantial differences between the two groups in these variables. The first row, “distance” represents the summary statistics of the distance measure, which is the propensity score between groups in this example.  
  
The second table shows the difference in each of the variables in the matched sample. The standardised mean differences of all variables are now much smaller and are very close to zero. Analyses using the matched sample are then unlikely to be confounded by these variables.

The last table indicates that every smoker (N = 974) is matched to a non-smoker (N = 974).  
  
These differences can be visualised using the following codes.

plot(summary(smk\_matching), abs = FALSE)

This produces the plot below.  


**Estimating the treatment effect**

We can now use the following codes to extract the matched data for further analysis.

matched\_data <- match.data(smk\_matching)

The association between psychological distress and smokers can now be tested using linear regression with the matched data. For individuals that are matched to another individual, they will have a weight of 1 in the dataset, for those who are not matched, they will have a weight of 0. Therefore, we will need to specify the weight in this analysis.

smk\_model1 <- lm(psyc\_distress ~ smoker, data = matched\_data, weights = weights)  
summary(smk\_model1)

This produces the following results.

Call:  
lm(formula = psyc\_distress ~ smoker, data = matched\_data, weights = weights)

Residuals:  
 Min 1Q Median 3Q Max   
-7.313 -5.313 -2.313 2.687 34.406

Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 15.5945 0.2203 70.797 < 2e-16 \*\*\*  
smoker 1.7187 0.3115 5.517 3.9e-08 \*\*\*

---  
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.874 on 1946 degrees of freedom  
Multiple R-squared: 0.0154, Adjusted R-squared: 0.0149

F-statistic: 30.44 on 1 and 1946 DF, p-value: 3.902e-08

However, it should be noted that for inference, we will need to calculate the cluster-robust standard error and the corresponding confidence interval (e.g. 95% confidence interval) using the codes below.

coeftest(smk\_model1, vcov. = vcovCL, cluster = ~subclass)  
coefci(smk\_model1, vcov. = vcovCL, cluster = ~subclass, level = 0.95)

The produces the following results. The standard errors are now properly adjusted for the matched sample design.

t test of coefficients:

Estimate Std. Error t value Pr(>|t|)   
(Intercept) 15.59446 0.19463 80.1219 < 2e-16 \*\*\*  
**smoker 1.71869 0.31540 5.4493 5.7e-08 \*\*\***

---  
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  
  
 2.5 % 97.5 %  
(Intercept) 15.212743 15.976169  
**smoker 1.100133 2.337239**

The above analysis shows that among smokers, psychological distress is 1.72 (95% CI [1.10, 2.34]) points higher than what would have been if they did not smoke.

**Doubly robust estimation**

Further, covariates can be included in the regression model to obtain a doubly robust estimate. This estimate is doubly robust because it only requires one model, either the model for estimating the propensity score or the outcome regression model, to be correct to yield an unbiased estimate.

It should be noted that when doubly robust estimation is used, coefficients from variables other than the target exposure should not be interpreted, even if they are “statistically significant”.  
  
smk\_model2 <- lm(psyc\_distress ~ smoker + sex + indigeneity + high\_school + partnered + remoteness + language + risky\_alcohol + age, data = matched\_data, weights = weights)

summary(smk\_model2)  
coeftest(smk\_model2, vcov. = vcovCL, cluster = ~subclass)  
coefci(smk\_model2, vcov. = vcovCL, cluster = ~subclass, level = 0.95)

These codes produce the results below.

Call:  
lm(formula = psyc\_distress ~ smoker + sex + indigeneity + high\_school +   
 partnered + remoteness + language + risky\_alcohol + age,   
 data = matched\_data, weights = weights)

Residuals:  
 Min 1Q Median 3Q Max   
-10.263 -4.563 -1.797 2.713 34.221

Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 20.83717 1.03325 20.167 < 2e-16 \*\*\*  
smoker 1.74366 0.30297 5.755 1.00e-08 \*\*\*  
sex -0.86217 0.31017 -2.780 0.00549 \*\*   
indigeneity 0.82341 0.68550 1.201 0.22982   
high\_school -0.01632 0.32486 -0.050 0.95994   
partnered -2.35743 0.30621 -7.699 2.18e-14 \*\*\*  
remoteness1 -0.07827 0.37835 -0.207 0.83613   
remoteness2 -0.78967 0.40159 -1.966 0.04940 \*   
language 0.57965 0.78788 0.736 0.46200   
risky\_alcohol 0.01953 0.32400 0.060 0.95194   
age -0.08103 0.01177 -6.883 7.87e-12 \*\*\*  
---  
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  
  
Residual standard error: 6.677 on 1937 degrees of freedom  
Multiple R-squared: 0.07543, Adjusted R-squared: 0.07065   
F-statistic: 15.8 on 10 and 1937 DF, p-value: < 2.2e-16

t test of coefficients:

Estimate Std. Error t value Pr(>|t|)   
(Intercept) 20.837167 1.001306 20.8100 < 2.2e-16 \*\*\*  
**smoker 1.743656 0.309151 5.6401 1.949e-08 \*\*\***  
sex -0.862165 0.298389 -2.8894 0.003903 \*\*   
indigeneity 0.823414 0.681772 1.2078 0.227289   
high\_school -0.016318 0.334055 -0.0488 0.961046   
partnered -2.357429 0.297713 -7.9185 4.015e-15 \*\*\*  
remoteness1 -0.078273 0.377376 -0.2074 0.835709   
remoteness2 -0.789671 0.386109 -2.0452 0.040970 \*   
language 0.579646 0.778095 0.7450 0.456389   
risky\_alcohol 0.019528 0.329395 0.0593 0.952731   
age -0.081031 0.011188 -7.2423 6.334e-13 \*\*\*  
---  
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

2.5 % 97.5 %  
(Intercept) 18.8734152 22.80091790  
**smoker 1.1373524 2.34995902**  
sex -1.4473624 -0.27696833  
indigeneity -0.5136696 2.16049671  
high\_school -0.6714630 0.63882722  
partnered -2.9412999 -1.77355736  
remoteness1 -0.8183786 0.66183339  
remoteness2 -1.5469041 -0.03243845  
language -0.9463465 2.10563879  
risky\_alcohol -0.6264776 0.66553412  
age -0.1029735 -0.05908804

This analysis, using doubly robust estimation, shows that among smokers, psychological distress is 1.74 (95% CI [1.13, 2.35]) points higher than what would have been if they were non-smokers.

For binary outcomes, logistic regression can be used to estimate the treatment effect. The outcome variable can be regressed on the treatment variable alone to obtain the treatment effect (e.g. ATT). However, application of the doubly robust method will be more difficult. If we simply add variables into the logistic regression, the estimate (beta coefficient or odd ratio) for the treatment represents a conditional effect instead of a marginal effect.

# Appendix 2. Inverse probability treatment weighting (cross-sectional/ pre-post design)

**Loading libraries and dataset**

In addition to the libraries (e.g., tidyverse) used in the example in Appendix 1, we will need the twang library. It can be installed using the codes.

install.packages("twang", dependencies = TRUE)  
install.packages("survey", dependencies = TRUE)

After installing the library, it can be loaded with the following lines (You will also need to load the libraries from Appendix 1).

library("twang")  
library("survey")

The same dataset will be used, and it can be loaded with the code

smk\_data <- read\_csv("https://raw.githubusercontent.com/gckc123/Causal\_Analysis\_Addiction\_Examples/main/smoking\_psyc\_distress.csv")

Explanations of the variables can be found in Appendix 1.

Since remoteness is a categorical variable with 3 levels, we first need to convert it into a factor variable.

smk\_data$remoteness <- as.factor(smk\_data$remoteness)

**Weighting and evaluating balance between treatment and control group**

In this example, we will demonstrate estimating the *average treatment effect* (ATE) by specifying the *estimand* parameter as “ATE”. This can be changed to “ATT” if the researcher is interested in average treatment effect among the treated. The setting of other parameters are generally sufficient for most scenarios. Interested readers can use the code “?ps” to invoke the help page for the ps function for more detailed explanations.

To create a pseudo-population in which the treatment and control group are balanced, the following codes can be used to calculate the IPTW.

smk\_iptw <- ps(smoker ~ sex + indigeneity + high\_school + partnered + remoteness + language + risky\_alcohol + age, interaction.depth = 3, data = as.data.frame(smk\_data), n.tree = 10000, estimand = "ATE", verbose = FALSE)

We can then use the plot() function to check the convergence of the algorithm, and the bal.table() function to check the balance between the treatment and control group.

plot(smk\_iptw)  
bal.table(smk\_iptw)

The plot function produces the following plot. The algorithm converges and remains stable.

A picture containing chart

Description automatically generated

The bal.table() function produces the following summary. We will focus on the first and last table. The first table shows the difference between the treatment and control group in the original unweighted data. The last table shows the difference in the weighted data. Similar to the matching method above, there are substantial differences in all variables in the original data, with all standardised effect size differences above 0.1. In the weighted data, a much better balance is achieved with all standardised effect size differences close to 0.

$unw

tx.mn tx.sd ct.mn ct.sd **std.eff.sz** stat pval ks ks.pval  
sex 0.494 0.500 0.442 0.497 **0.104** 3.031 0.002 0.052 0.020  
indigeneity 0.052 0.223 0.018 0.131 **0.239** 4.770 0.000 0.035 0.250  
high\_school 0.422 0.494 0.638 0.481 **-0.443** -12.820 0.000 0.216 0.000  
partnered 0.463 0.499 0.691 0.462 **-0.483** -13.504 0.000 0.228 0.000  
remoteness:0 0.585 0.493 0.677 0.467 **-0.195**  19.793 0.000 0.092 0.000  
remoteness:1 0.218 0.413 0.190 0.392  **0.070**  NA NA 0.028 NA  
remoteness:2 0.197 0.398 0.133 0.339  **0.186**  NA NA 0.064 NA  
language 0.958 0.201 0.913 0.282  **0.164**  6.180 0.000 0.045 0.064  
risky\_alcohol 0.643 0.479 0.541 0.498  **0.204**  6.169 0.000 0.102 0.000  
age 51.606 12.990 53.782 14.333 **-0.153**  -4.839 0.000 0.102 0.000

$ks.mean.ATE  
 tx.mn tx.sd ct.mn ct.sd std.eff.sz stat pval ks ks.pval  
sex 0.449 0.498 0.448 0.497 0.002 0.051 0.959 0.001 1.000  
indigeneity 0.019 0.138 0.021 0.144 -0.012 -0.485 0.627 0.002 1.000  
high\_school 0.608 0.488 0.612 0.487 -0.009 -0.217 0.828 0.004 1.000  
partnered 0.651 0.477 0.664 0.472 -0.028 -0.734 0.463 0.013 1.000  
remoteness:0 0.668 0.471 0.667 0.471 0.003 0.003 0.997 0.001 0.997  
remoteness:1 0.193 0.395 0.193 0.395 -0.001 NA NA 0.000 NA  
remoteness:2 0.139 0.346 0.140 0.347 -0.003 NA NA 0.001 NA  
language 0.929 0.257 0.918 0.274 0.039 0.847 0.397 0.011 1.000  
risky\_alcohol 0.572 0.495 0.554 0.497 0.037 0.862 0.389 0.018 0.992  
age 53.293 13.857 53.525 14.194 -0.016 -0.373 0.709 0.016 0.999

$es.mean.ATE  
 tx.mn tx.sd ct.mn ct.sd std.eff.sz stat pval ks ks.pval  
sex 0.449 0.498 0.448 0.497  **0.002**  0.056 0.955 0.001 1.000  
indigeneity 0.020 0.138 0.021 0.144  **-0.012** -0.467 0.640 0.002 1.000  
high\_school 0.608 0.488 0.612 0.487  **-0.008** -0.215 0.830 0.004 1.000  
partnered 0.650 0.477 0.664 0.472  **-0.029** -0.757 0.449 0.014 1.000  
remoteness:0 0.667 0.471 0.667 0.471  **0.002**  0.003 0.997 0.001 0.997  
remoteness:1 0.193 0.395 0.193 0.395  **0.000**  NA NA 0.000 NA  
remoteness:2 0.139 0.346 0.140 0.347  **-0.003**  NA NA 0.001 NA  
language 0.929 0.257 0.918 0.274  **0.039**  0.847 0.397 0.011 1.000  
risky\_alcohol 0.572 0.495 0.554 0.497  **0.037**  0.870 0.384 0.018 0.991  
age 53.296 13.859 53.525 14.194  **-0.016** -0.368 0.713 0.016 0.999

**Estimating the treatment effect**

We can now extract the weight to perform a weighted analysis. The following code will extract the weight from the previous step and save it into a new column “weight” in the data.

smk\_data$weight <- get.weights(smk\_iptw, stop.method = "es.mean")

We can then use the svydesign() function from the survey package to set up the design of the analysis and the svyglm() to estimate the effect.

design\_iptw <- svydesign(ids = ~1, weights = ~weight, data = smk\_data)  
smk\_model3 <- svyglm(psyc\_distress ~ smoker, design = design\_iptw)  
summary(smk\_model3)  
confint(smk\_model3)

The above codes produce the following results.

Call:  
svyglm(formula = psyc\_distress ~ smoker, design = design\_iptw)

Survey design:  
svydesign(ids = ~1, weights = ~weight, data = smk\_data)

Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 14.83600 0.06618 224.168 < 2e-16 \*\*\*  
**smoker 1.72865 0.25771 6.708 2.11e-11 \*\*\***  
---  
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 38.62049)  
Number of Fisher Scoring iterations: 2

2.5 % 97.5 %  
(Intercept) 14.706263 14.965732  
**smoker 1.223468 2.233829**

This analysis estimates an increase of 1.73 (95% [1.22, 2.23]) points in psychological distress among smokers compared to non-smokers.

**Doubly robust estimation**

Similar to the matching example in Appendix 1, we can add covariates into the outcome model to obtain a doubly robust estimate.

smk\_model4 <- svyglm(psyc\_distress ~ smoker + sex + indigeneity + high\_school + partnered + remoteness + language + risky\_alcohol + age, design = design\_iptw)  
summary(smk\_model4)  
confint(smk\_model4)

The above codes produce the following results.

Call:  
svyglm(formula = psyc\_distress ~ smoker + sex + indigeneity +   
 high\_school + partnered + remoteness + language + risky\_alcohol +   
 age, design = design\_iptw)

Survey design:  
svydesign(ids = ~1, weights = ~weight, data = smk\_data)

Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 19.904285 0.715860 27.805 < 2e-16 \*\*\*  
**smoker 1.671872 0.247822 6.746 1.62e-11 \*\*\***  
sex -0.757545 0.241068 -3.142 0.00168 \*\*   
indigeneity 0.841129 0.569268 1.478 0.13956   
high\_school -0.145922 0.252175 -0.579 0.56284   
partnered -2.075947 0.250193 -8.297 < 2e-16 \*\*\*  
remoteness1 0.084483 0.334046 0.253 0.80035   
remoteness2 -0.763333 0.285282 -2.676 0.00747 \*\*   
language 1.417293 0.467557 3.031 0.00244 \*\*   
risky\_alcohol -0.233845 0.252525 -0.926 0.35446   
age -0.081472 0.008858 -9.197 < 2e-16 \*\*\*  
---  
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 36.15502)  
Number of Fisher Scoring iterations: 2

2.5 % 97.5 %  
(Intercept) 18.50101360 21.30755703  
**smoker 1.18607503 2.15766846**  
sex -1.23010174 -0.28498765  
indigeneity -0.27478393 1.95704236  
high\_school -0.64025121 0.34840649  
partnered -2.56639078 -1.58550318  
remoteness1 -0.57033398 0.73930040  
remoteness2 -1.32256048 -0.20410520  
language 0.50075865 2.33382761  
risky\_alcohol -0.72886034 0.26117022  
age -0.09883716 -0.06410729

This analysis, using doubly robust estimation, estimates an increase of 1.67 points (95% CI [1.18, 2.16]) in psychological distress among smokers compared to non-smokers.

# Appendix 3. Inverse probability treatment weighting (multi-wave longitudinal design)

In this example, we will use a simulated dataset to test the impact of parental alcohol supply during adolescence on risky alcohol use in young adulthood.

Similar to the example above (Appendix 2), the tidyverse, twang and survey library are needed.

library("tidyverse")  
library("twang")  
library("survey")

**Load the dataset**  
The following codes can be used to load the dataset.

alc\_data <- read\_csv("https://raw.githubusercontent.com/gckc123/Causal\_Analysis\_Addiction\_Examples/main/home\_alc.csv")

This dataset contains 18 variables.

**sex:** 0: Female; 1: Male **alc\_use1 – alc\_use4:** Alcohol use at wave 1 to 4 during adolescence; 0 – no alcohol use; 1 – alcohol use  
**smk1 – smk4:** Smoking at wave 1 to 4; 0 – no smoking; 1 - smoking  
**adult\_alc\_risky**: Risky alcohol use in wave 5 (Young adulthood); 0 – no risky alcohol use; 1 – engaged risky alcohol use  
**home\_alc\_1 – home\_alc\_4**: Access to alcohol at home at wave 1 to 4: 0 – no access; 1 – had access  
**alc\_peer1 – alc\_peer4:** Having alcohol using peer at wave 1 to 4: 0 – no alcohol using peer; 1 – had alcohol using peers

**Weighting between treatment and control group**

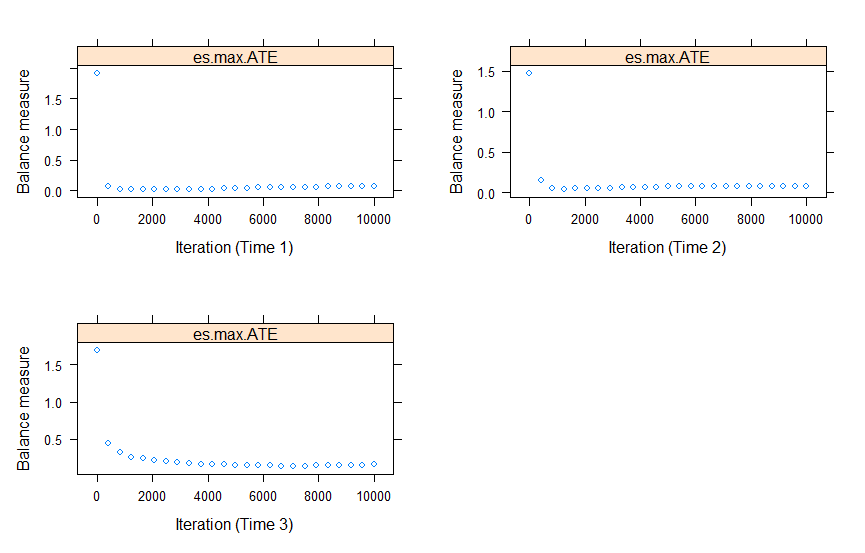
This is a simulated longitudinal dataset on alcohol access at home during adolescence and future risky alcohol use. It has five waves of data, wave 1 to 4 measures whether an adolescent has access to alcohol at home and other alcohol related variables. Wave 5 measures risky alcohol consumption in early adulthood.  
  
In this analysis, we will estimate the effect of number of waves the adolescent reported having home alcohol access between wave 2 and 4 on future risky alcohol use in wave 5 (young adulthood). Data from wave 1 will be treated as baseline.  
  
The iptw() function in the twang package can be used to calculate the weight. The first parameter is a list of formulas to specify the exposure in each wave (home alcohol access: Yes/ No), and variables that are to be used in the weighting procedure. The parameter cumulative and priorTreatment are both set to TRUE. Therefore, the weight for home alcohol access at wave 3 and 4 will be calculated based on the variables specified in the formula for wave 3 and 4, previous exposure (home alcohol access at wave 2 and 3 respectively) and also variables specified in the formula for wave 2 and wave 3. We also specify sex to be a time invariant variable by setting timeInvariant ~ sex, so that the variable sex will be included in the weight calculation in all waves.

alc\_iptw <- iptw(list(home\_alc\_2 ~ home\_alc\_1 + alc\_use1 + alc\_peer1 + smk1,  
 home\_alc\_3 ~ alc\_use2 + alc\_peer2 + smk2,  
 home\_alc\_4 ~ alc\_use3 + alc\_peer3 + smk3  
 ),  
 timeInvariant ~ sex,  
 data = as.data.frame(alc\_data),  
 cumulative = TRUE,  
 priorTreatment = TRUE,  
 stop.method = "es.max",  
 n.trees = 10000)

The plot() function can be used to access convergence.

plot(alc\_iptw, plots = 1)

This line produces the following plot, which indicates that the algorithm converges and remains stable for all three waves.



**Evaluating balance**

Similar to the IPTW example in Appendix 2, the bal.table() function can be used to evaluate balance between group.

bal.table(alc\_iptw)

The bal.table() function estimates the standardized effect size in the unweighted and weighted sample in each wave.

Balance at time 1 :

$unw  
 tx.mn tx.sd ct.mn ct.sd std.eff.sz stat p ks ks.pval  
home\_alc\_1 0.418 0.496 0.028 0.165 **1.810** 6.997 0.000 0.390 0.000  
alc\_use1 0.253 0.438 0.094 0.291 **0.527** 3.221 0.001 0.160 0.044  
alc\_peer1 0.456 0.501 0.278 0.448 **0.393** 3.102 0.002 0.178 0.018  
smk1 0.443 0.500 0.264 0.441 **0.402** 3.136 0.002 0.179 0.016  
sex 0.443 0.500 0.538 0.499 **-0.190** -1.647 0.100 0.095 0.514

$es.max.ATE  
 tx.mn tx.sd ct.mn ct.sd std.eff.sz stat p ks ks.pval  
home\_alc\_1 0.050 0.219 0.048 0.214 **0.008** 0.123 0.902 0.002 1  
alc\_use1 0.100 0.302 0.102 0.302 **-0.004** -0.037 0.970 0.001 1  
alc\_peer1 0.277 0.450 0.286 0.452 **-0.021** -0.147 0.883 0.010 1  
smk1 0.265 0.444 0.273 0.446 **-0.018** -0.123 0.902 0.008 1  
sex 0.521 0.503 0.533 0.499 **-0.023** -0.147 0.883 0.011 1  
  
Balance at time 2 :  
$unw  
 tx.mn tx.sd ct.mn ct.sd std.eff.sz stat p ks ks.pval  
home\_alc\_1 0.289 0.456 0.029 0.167 **1.211** 6.099 0.000 0.261 0.000  
alc\_use1 0.228 0.421 0.092 0.289 **0.451** 3.405 0.001 0.136 0.040  
alc\_peer1 0.553 0.499 0.266 0.442 **0.634** 5.973 0.000 0.287 0.000  
smk1 0.465 0.501 0.258 0.437 **0.465** 4.303 0.000 0.207 0.000  
sex 0.482 0.502 0.537 0.499 **-0.109** -1.116 0.265 0.054 0.915  
alc\_use2 0.298 0.460 0.119 0.324 **0.528** 4.097 0.000 0.179 0.002  
alc\_peer2 0.649 0.479 0.377 0.485 **0.555** 5.836 0.000 0.272 0.000  
smk2 0.404 0.493 0.278 0.448 **0.276** 2.631 0.009 0.125 0.074  
home\_alc\_2 0.377 0.487 0.026 0.159 **1.572** 7.700 0.000 0.351 0.000

$es.max.ATE  
 tx.mn tx.sd ct.mn ct.sd std.eff.sz stat p ks ks.pval  
home\_alc\_1 0.052 0.222 0.046 0.209 **0.028** 0.410 0.682 0.006 1  
alc\_use1 0.110 0.315 0.101 0.302 **0.029** 0.279 0.780 0.009 1  
alc\_peer1 0.312 0.465 0.285 0.452 **0.061** 0.513 0.608 0.027 1  
smk1 0.314 0.466 0.273 0.446 **0.091** 0.711 0.477 0.041 1  
sex 0.493 0.502 0.533 0.499 **-0.078** -0.567 0.571 0.039 1  
alc\_use2 0.143 0.352 0.133 0.339 **0.031** 0.298 0.766 0.011 1  
alc\_peer2 0.420 0.496 0.395 0.489 **0.051** 0.386 0.700 0.025 1  
smk2 0.247 0.433 0.289 0.453 **-0.092** -0.825 0.410 0.042 1  
home\_alc\_2 0.059 0.236 0.049 0.215 **0.045** 0.685 0.493 0.010 1

Balance at time 3 :

$unw  
 tx.mn tx.sd ct.mn ct.sd std.eff.sz stat p ks ks.pval  
home\_alc\_1 0.260 0.441 0.037 0.189 **1.034** 4.429 0.000 0.222 0.001  
alc\_use1 0.234 0.426 0.095 0.293 **0.459** 2.842 0.005 0.139 0.119  
alc\_peer1 0.468 0.502 0.278 0.448 **0.420** 3.269 0.001 0.190 0.010  
smk1 0.364 0.484 0.268 0.443 **0.214** 1.697 0.090 0.095 0.522  
sex 0.351 0.480 0.543 0.498 **-0.384** -3.428 0.001 0.192 0.009  
alc\_use2 0.312 0.466 0.123 0.329 **0.556** 3.526 0.000 0.189 0.011  
alc\_peer2 0.649 0.480 0.384 0.487 **0.541** 4.739 0.000 0.265 0.000  
smk2 0.338 0.476 0.285 0.452 **0.116** 0.948 0.343 0.052 0.988  
home\_alc\_2 0.390 0.491 0.034 0.182 **1.590** 6.365 0.000 0.355 0.000  
alc\_use3 0.325 0.471 0.155 0.362 **0.457** 3.123 0.002 0.169 0.030  
alc\_peer3 0.649 0.480 0.479 0.500 **0.342** 3.050 0.002 0.171 0.028  
smk3 0.403 0.494 0.281 0.450 **0.268** 2.125 0.034 0.122 0.231  
home\_alc\_3 0.532 0.502 0.051 0.221 **1.815** 8.415 0.000 0.481 0.000

$es.max.ATE  
 tx.mn tx.sd ct.mn ct.sd std.eff.sz stat p ks ks.pval  
home\_alc\_1 0.038 0.193 0.045 0.207 **-0.032** -0.448 0.654 0.007 1  
alc\_use1 0.087 0.283 0.100 0.300 **-0.043** -0.311 0.756 0.013 1  
alc\_peer1 0.313 0.467 0.284 0.451 **0.064** 0.280 0.780 0.029 1  
smk1 0.257 0.440 0.270 0.444 **-0.031** -0.138 0.891 0.014 1  
sex 0.528 0.503 0.536 0.499 **-0.017** -0.072 0.942 0.008 1  
alc\_use2 0.113 0.318 0.130 0.336 **-0.050** -0.369 0.713 0.017 1  
alc\_peer2 0.445 0.500 0.393 0.489 **0.107** 0.454 0.650 0.052 1  
smk2 0.266 0.445 0.287 0.453 **-0.045** -0.206 0.837 0.021 1  
home\_alc\_2 0.049 0.218 0.046 0.208 **0.017** 0.233 0.816 0.004 1  
alc\_use3 0.137 0.347 0.161 0.367 **-0.063** -0.429 0.668 0.023 1  
alc\_peer3 0.462 0.502 0.485 0.500 **-0.045** -0.193 0.847 0.022 1  
smk3 0.238 0.429 0.286 0.452 **-0.107** -0.575 0.565 0.048 1  
home\_alc\_3 0.079 0.272 0.067 0.251 **0.045** 0.469 0.639 0.012 1

The weighted sample is much more balanced between the treatment and control groups.

**Estimating the treatment effect**

After evaluating balance, we can now extract the unstabilised weight into a new column unstab\_weight in the dataset.

alc\_data$unstab\_weight <- get.weights.unstab(alc\_iptw, stop.method = "es.mean")[,1]

To calculate the stabilised weight, we will need to estimate the unconditional probability of home alcohol access at wave 2, and the probability of home alcohol access in subsequence waves conditional on home alcohol access in previous waves. To do this, we will need to run three logistic regression using the glm() function.

num\_fm <- list(glm(home\_alc\_2 ~ 1, family = binomial, data = alc\_data),  
 glm(home\_alc\_3 ~ home\_alc\_2, family = binomial, data = alc\_data),  
 glm(home\_alc\_4 ~ home\_alc\_2 + home\_alc\_3, family = binomial, data = alc\_data))

We then use the get.weights.num() function to calculate the numerators of the stabilised weight.  
  
num\_weights <- get.weights.num(alc\_iptw, num\_fm)

Lastly, we multiply the numerators to the unstablised weight to obtain the stablised weight.  
  
alc\_data$stab\_weight <- num\_weights\* alc\_data$unstab\_weight

Before estimating the effect of home alcohol access, we create a variable that represents the total number of waves the individual reported having home alcohol access. In this analysis, we treat this as a categorical variable with four level (0, 1, 2 and 3). The function as.factor() is used to convert it from a numeric variable to factor variable.

alc\_data$total\_home\_alc = alc\_data$home\_alc\_2 + alc\_data$home\_alc\_3 + alc\_data$home\_alc\_4   
alc\_data$total\_home\_alc = as.factor(alc\_data$total\_home\_alc)

We can now use the svydesign() function to setup the weighted analysis, and use the svyglm() function to conduct a weighted logistic regression analysis.

design\_iptw <- svydesign(ids = ~1, weights = ~stab\_weight, data = alc\_data)  
alc\_model <- svyglm(adult\_alc\_risky ~ total\_home\_alc, design = design\_iptw, family = binomial)  
summary(alc\_model)  
exp(coef(alc\_model))  
exp(confint(alc\_model))

The above codes produce the following outputs. The last two lines use the exp() function to calculate odds ratios from the model coefficients.  
  
Call:  
svyglm(formula = adult\_alc\_risky ~ total\_home\_alc, design = design\_iptw,   
 family = binomial)

Survey design:  
svydesign(ids = ~1, weights = ~stab\_weight, data = alc\_data)

Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) -1.50882 0.07511 -20.088 <2e-16 \*\*\*  
total\_home\_alc1 **0.40334** 0.27554 1.464 0.143   
total\_home\_alc2 **-0.55129** 0.54824 -1.006 0.315   
total\_home\_alc3 **1.05551** 0.72068 1.465 0.143   
---  
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1.000667)

Number of Fisher Scoring iterations: 4  
  
 (Intercept) total\_home\_alc1 total\_home\_alc2 total\_home\_alc3   
 **0.2211703 1.4968183 0.5762044 2.8734383**  
 2.5 % 97.5 %  
(Intercept) **0.1908713 0.256279**  
total\_home\_alc1 **0.8718510 2.569779**  
total\_home\_alc2 **0.1965796 1.688942**  
total\_home\_alc3 **0.6989714 11.812568**

This analysis did not provide any evidence that home alcohol access during adolescence increases risky alcohol use in young adulthood, because all the results are completely not statistically significant and the estimates are not consistent with odds ratios pointing to different directions depending on the number of waves the adolescent had access to alcohol.

# Appendix 4. Interrupted time series analysis with control

In this example, we will estimate the effect of the minimum alcohol pricing in Northern Territory, Australia, on alcohol consumption at the population level.

**Install and load the necessary R library**

In addition to the libraries used in the previous example (see the required libraries in Appendix 1-3), we will need the nlme and ggplot2 libraries. It can be installed using the codes.

install.packages("nlme", dependencies = TRUE)  
install.packages("ggplot2", dependencies = TRUE)

After installation, the library can be loaded using the following line.  
  
library(nlme)  
library(ggplot2)

**Load the dataset**

alc\_mup\_data <- read\_csv("https://statsnotebook.io/blog/data\_management/example\_data/alcohol\_data\_NTWA.csv")

This dataset contains 5 variables  
**alcohol**: measure of population level alcohol consumption in a month.   
**time:** time measures in months  
**state:** “NT”: Northern Territory; “WA” Western Australia  
**intervention:** Whether the time point is pre or post intervention. 0: Pre-intervention; 1: Post-intervention.  
**season:** Season of the year. 1: Spring; 2: Summer; 3: Autumn; 4: Winter

For this analysis, we firstly convert the state and intervention variables into factor. We also use the relevel() function to set the reference level of the state variable to be “WA” (Western Australia).

alc\_mup\_data$state <- factor(alc\_mup\_data$state, exclude = c("", NA))  
alc\_mup\_data$intervention <- factor(alc\_mup\_data$intervention, exclude = c("", NA))  
  
#change the reference level to be “Western Australia”  
alc\_mup\_data$state <- relevel(alc\_mup\_data$state, ref="WA")

**Descriptive analysis**

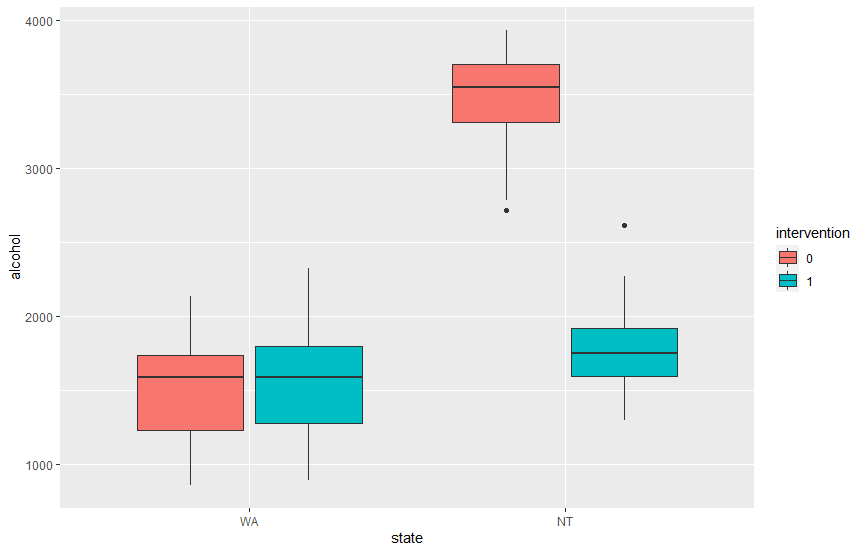
We can generate the descriptive statistics by states and by pre/post intervention.

alc\_mup\_data %>%  
 group\_by(state, intervention) %>%  
 summarize(count = n(),  
 M\_alcohol = mean(alcohol, na.rm = TRUE),  
 Mdn\_alcohol = median(alcohol, na.rm = TRUE),  
 SD\_alcohol = sd(alcohol, na.rm = TRUE),  
 IQR\_alcohol = IQR(alcohol, na.rm = TRUE)  
 ) %>%   
 print()

The above codes produce the following outputs.  
  
 state intervention count M\_alcohol Mdn\_alcohol SD\_alcohol IQR\_alcohol  
 <fct> <fct> <int> <dbl> <dbl> <dbl> <dbl>  
1 NT 0 24 3444. 3548. 349. 388.  
2 NT 1 24 1777. 1748. 323. 324.  
3 WA 0 24 1531. 1587. 358. 506.  
4 WA 1 24 1582. 1583. 367. 517.

We can also use the ggplot() function to visualise alcohol consumption level by state and by pre/post intervention.

ggplot(alc\_mup\_data) +  
 geom\_boxplot(aes(y=alcohol, x=state, fill = intervention))  
  
The above codes produce the following boxplot.



The descriptive analysis suggests that alcohol minimum pricing likely reduced alcohol consumption in the population.

**Estimating the intervention effect**

We now use the gls() function to test the effect of alcohol minimum pricing on population level alcohol consumption. We use the correlation parameter to specify the auto-correlation structure. Here, we specify a lag-1 correlation. For the form parameter, we will need to specify the variable representing the elapse of time and the variable indicating whether the data is from the intervention group or the control group (state in this example).

res <- gls(alcohol ~ time\*intervention\*state,  
 data = alc\_mup\_data,  
 correlation = corARMA(p = 1, form =~ time | state), method = "ML")  
summary(res)

The above codes produce the following outputs.

Generalized least squares fit by maximum likelihood  
 Model: alcohol ~ time \* intervention \* state   
 Data: alc\_mup\_data   
 AIC BIC logLik  
 1330.261 1355.905 -655.1307  
  
Correlation Structure: AR(1)  
 Formula: ~time | state   
 Parameter estimate(s):  
 Phi   
0.7066424

Coefficients:  
 Value Std.Error t-value p-value  
(Intercept) 1558.506 263.2032 5.921303 0.0000  
time 1.636 16.9157 0.096720 0.9232  
intervention1 -20.840 715.1188 -0.029143 0.9768  
stateNT 2133.915 372.2256 5.732854 0.0000  
time:intervention1 -1.221 27.5750 -0.044281 0.9648  
time:stateNT -23.652 23.9224 -0.988718 0.3255  
intervention1:stateNT -3258.232 1011.3307 -3.221728 0.0018  
time:intervention1:stateNT 62.170 38.9969 1.594227 0.1145

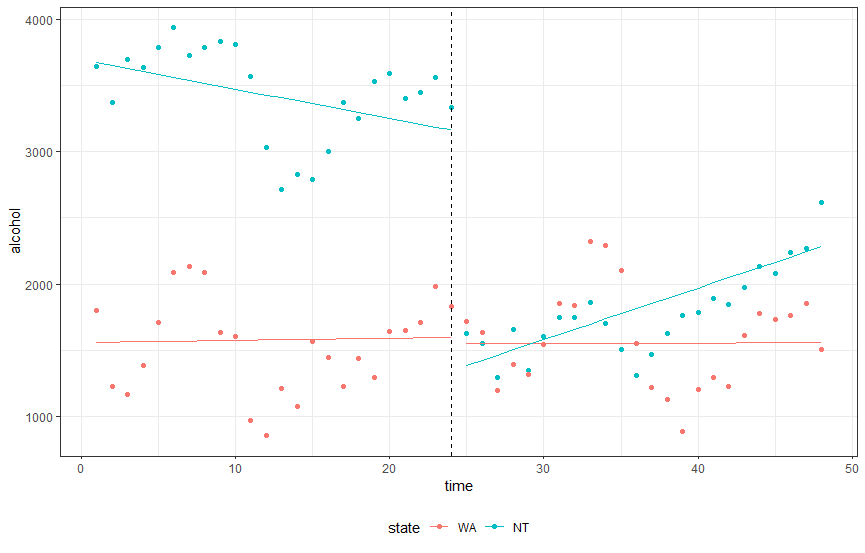
Correlation:   
 (Intr) time intrv1 statNT tm:nt1 tm:sNT in1:NT  
time -0.840   
intervention1 -0.507 0.639   
stateNT -0.707 0.594 0.358   
time:intervention1 0.600 -0.815 -0.945 -0.424   
time:stateNT 0.594 -0.707 -0.452 -0.840 0.576   
intervention1:stateNT 0.358 -0.452 -0.707 -0.507 0.668 0.639 time:intervention1:stateNT -0.424 0.576 0.668 0.600 -0.707 -0.815 -0.945

Standardized residuals:  
 Min Q1 Med Q3 Max   
-2.3167349 -0.8427548 0.0837754 0.6497867 2.4752763

Residual standard error: 312.3269   
Degrees of freedom: 96 total; 88 residual  
  
There is a strong significant intervention by state interaction. To facilitate interpretation, we visualise the results using the following codes.  
  
#generating the model-based prediction  
alc\_mup\_data$predicted <- res$fitted  
  
#generating the interaction for ggplots  
groups = interaction(alc\_mup\_data$intervention,alc\_mup\_data$state)

#ploting the time series  
plot <- ggplot() +  
 geom\_point(data = alc\_mup\_data, aes(y = alcohol, x = time, color = state)) +  
 geom\_line(data = alc\_mup\_data, aes(y = predicted, x = time, color = state, group = groups)) +  
 geom\_vline(xintercept = max((alc\_mup\_data %>% filter(intervention == "0"))$time), linetype =   
"dashed") +  
 theme\_bw(base\_family = "sans") +  
 theme(legend.position = "bottom")  
plot

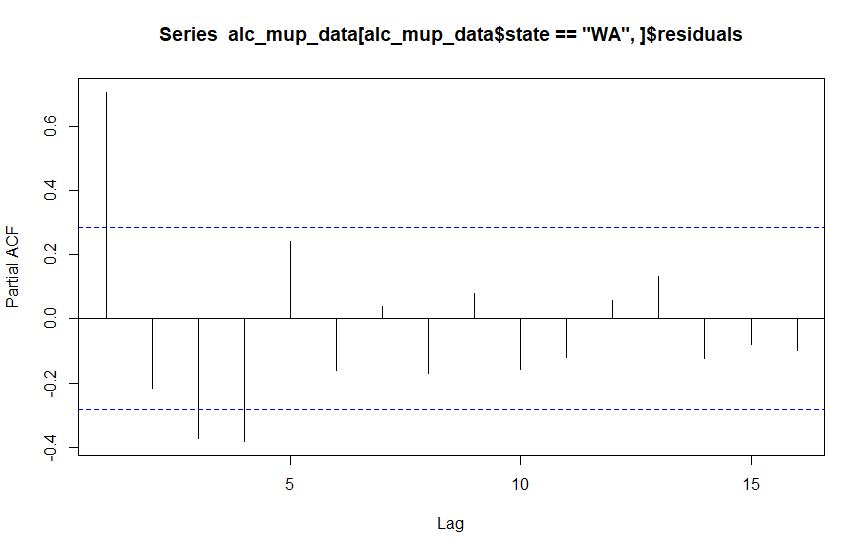
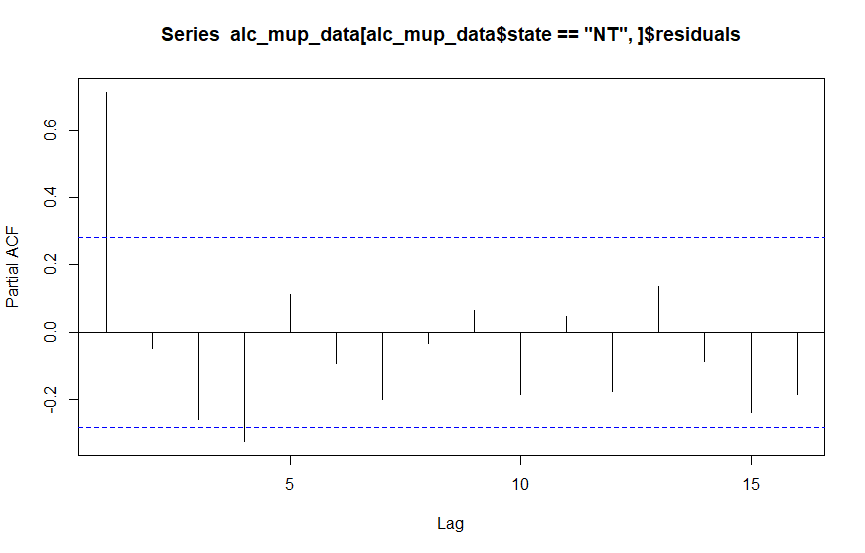
The above codes produce the following plot.



While alcohol consumption in the Northern Territory was higher than in Western Australia before alcohol minimum pricing, there was no significant difference in the pre-intervention trend (as indicated by the time by state interaction). Immediately after implementing minimum alcohol price, there was a significant drop in alcohol consumption in the Northern Territory but not in Western Australia (as indicated by the intervention by state interaction).

Our model has accounted for lag-1 autocorrelation. We can evaluate if this is sufficient using the partial auto-correlation function. The acf() can be used with the type parameter set to “partial”. In this analysis, we will need to plot the auto-correlation function for Western Australia and Northern Territory separately.

#extract the residuals  
alc\_mup\_data$residuals <- residuals(res)

#plotting the partial ACF  
acf(alc\_mup\_data[alc\_mup\_data$state == "NT",]$residuals, type = "partial")  
acf(alc\_mup\_data[alc\_mup\_data$state == "WA",]$residuals, type = "partial")  
  
  
  
Both PACF plots suggest that we should adjust for up to the lag-4 auto-correlation. The analysis can be rerun by setting the p parameter to 4, as follow.

res2 <- gls(alcohol ~ time\*intervention\*state,  
 data = alc\_mup\_data,  
 correlation = corARMA(**p = 4**, form =~ time | state), method = "ML")  
summary(res2)

**Adjusting for seasonality**  
For most public health interventions, the simplest way to adjust for seasonality is to include an extra variable that represents different seasons of the year. Alternatively, we can include a smooth harmonic function to account for the seasonal effect. The following codes can be used to adjust for seasonal effect with a smooth harmonic function. The library tsModel is needed. It can be installed and loaded using the codes below.  
  
#install the tsModel library  
install.packages("tsModel", dependencies = TRUE)

#load the nlme and tsModel libraries  
library(nlme)  
library(tsModel)

#The harmonic function is used to calculate the harmonic terms calculate based on sine and cosine function  
#the first parameter of this function specifies the time variable  
#the second specifies the number of sine and cosine pairs to include  
#the third specifies the length of the period  
  
alc\_mup\_data <- cbind(alc\_mup\_data, data.frame(harmonic(alc\_mup\_data$time, 1, 12)))  
alc\_mup\_data <- alc\_mup\_data %>%   
 rename(harmonic1 = X1,  
 harmonic2 = X2)

res <- gls(alcohol ~ time\*intervention\*state + harmonic1 + harmonic2,  
 data = alc\_mup\_data,  
 correlation = corARMA(p = 1, form =~ time | state), method = "ML")  
summary(res)  
  
#calcluate the predicted value  
alc\_mup\_data$predicted <- res$fitted  
groups = interaction(alc\_mup\_data$intervention,alc\_mup\_data$state)

#calculate the predicted linear trend for data visualisation  
alc\_mup\_data.linear <- alc\_mup\_data  
alc\_mup\_data.linear$harmonic1 <- 0  
alc\_mup\_data.linear$harmonic2 <- 0

alc\_mup\_data.linear$predicted <- predict(res, alc\_mup\_data.linear)  
  
#plot the time series  
plot <- ggplot() +

geom\_point(data = alc\_mup\_data, aes(y = alcohol, x = time, color = state)) +  
 geom\_line(data = alc\_mup\_data, aes(y = predicted, x = time, color = state, group = groups), linetype = "dashed") +  
 geom\_line(data = alc\_mup\_data.linear, aes(y = predicted, x = time, color = state, group = groups)) +  
 geom\_vline(xintercept = max((alc\_mup\_data %>% filter(intervention == "0"))$time), linetype = "dashed") +  
 theme\_bw(base\_family = "sans") +  
 theme(legend.position = "bottom")

plot

The above codes produce the following plot.

Chart, scatter chart

Description automatically generated

# Ending remarks

We hope you find this Appendix of explanations and R codes helpful.

All of these above analyses can also be run using StatsNotebook for R, a free downloadable software that provides a point-and-click interface for running analyses in R.

<https://statsnotebook.io/>

All the best with your analyses!