Difference-in-Difference with Various Treatment Timings and Multiple Periods

1. Introduction

The Difference-in-Difference (DiD) design of Callaway & Sant'Anna (2021) (CS (2021)) estimates the average treatment effects (ATE) in a scenario of staggered adoption, i.e., treated units stay treated in post-treatment periods. The proposed strategy is able to estimate well-identified ATE with multiple periods, while overcoming the arising challenges from variation of treatment timing and dynamic heterogeneous effects. In the following, the methodology of the CS (2021) is introduced and the problems associated with the dynamic two-way fixed effects (TWFE) regression are explained.

CS (2021) suggest a two-step procedure to estimate different aggregated average treatment effects. In advance, it is important to state that a group is defined by its first year of their treatment, i.e., a unit is treated in the year 2004 and, thus, belongs to group 2004. The first step of the procedure is to calculate the disaggregated ATE within groups, while the control group is either a "never-treated" group or a "not-yet-treated" group. This disaggregated estimator is referred to as the "group-time average treatment effect" ($ATT(g,t)$) and measures the effect of group g in the time period t. Thereby, this estimator represents the canonical DiD in the two-period case and can be interpreted as an average treatment effect of the treated. In the second step of the procedure those $ATT(g,t)$ can be aggregated to different overall ATE by weighting each $ATT(g,t)$ with the corresponding probability to be in group g and restricting the included $ATT(g,t)$ accordingly to the research question.

Furthermore, CS (2021) introduce a parallel trend assumption (PTA) that holds after conditioning on pre-treatment covariates. This represents an extension of the traditional PTA in the case that the treated and control group are not developing with the same trend in the absence of a treatment, due to covariate-specific trends of the groups. The conditional PTA solves this issue by controlling for these covariate-specific trends and allows a greater credibility of the DiD design through strengthening the PTA.

2. Unreliable Dynamic Two-Way Fixed Regressions

In this section, the original idea of the dynamic TWFE with staggered adoption is repeated with an accompanying explanation. Furthermore, the challenges of the dynamic TWFE are discussed as well as how the DiD design of CS (2021) solves these challenges.

The basic idea of a dynamic TWFE is to evaluate how the treatment effect of a policy evolves over several periods after its introduction. The basic dynamic TWFE can be extended for testing the PTA by allowing to include pre-treatment periods and let them interact with the treatment dummy. Lastly, the TWFE are included in order to capture everything what is affecting each unit equally over time and all the time-invariant characteristics of each unit.

The problems associated with the dynamic TWFE have been extensively discussed in the recent research. Thereby, it has been shown that the dynamic TWFE regression with staggered adoption and multiple groups lead to potentially biased estimators according to Goodman-Bacon (2021) and Chaisemartin & D'Haultfoeuille (2020). These authors are arguing that each estimator of the dynamic TWFE for each period is a weighted average of different smaller ATE. These smaller ATE are the canonical DiD estimators between a treated and control group and referred to as "sub-DiD estimators". To illustrate this finding, let's assume the setup with multiple periods and variation in treatment timing with a "not-yet-treated", "never-treated" group and the treated group in period t. The estimator of this treatment effect equals the weighted average of the sub-DiD estimators for the treated and not-yet-treated group and for the treated and never-treated group. This weighted average is associated with two problems that lead to a hard-to-interpret and biased estimator.

The first problem is described by Goodman-Bacon (2021) and is related to the composition of weights, because they depend on the squared sub-sample size and the sub-sample variance. Consequently, the so-called variance-weights are determined by the distance of the treatment timing to the middle of the time window, the distance in time between two treatments and the size of the time window. Therefore, the variance-weights can change significantly if one of these factors changes and as a result the magnitude of the coefficient is sensitive to changes of the time window.

The second problem is the so-called "negative weight problem", which is discussed by Chaisemartin & D'Haultfoeuille (2020). This problem arises in the case of multiple periods and multiple groups when an "already-treated" group is used as a control group in one of the dynamic TWFE estimators. For a better understanding, recall the aforementioned example on the composition of the dynamic TWFE estimator. In this example, the "not-yet-treated" and "never-treated" group were used as a control group for the TWFE-DiD-estimator, which are both valid control groups for the DiD design as both groups resemble a theoretical potential outcome of an untreated group (Callaway, 2023). On the otherside, the "already-treated" group is not a valid representation of the theoretical potential outcome of an untreated group since this group received a treatment at an earlier time point (Callaway, 2023). Now, if the "already-treated" group has dynamic treatment effects, i.e., the treatment effects do not stay constant over the post-treatment periods, then the sub-DiD estimator has negative weights attached (Callaway, 2023). Thus, the dynamic TWFE coefficient is biased and does not allow a reliable causal interpretation of the treatment (Callaway, 2023).

Overall, it can be concluded that the dynamic TWFE regression leads to biased estimators for each period in the case of dynamic heterogeneous effects, due to the two above mentioned reasons. CS (2021) can solve both of these problems by their two-step procedure to calculate the aggregated ATE by first calculating the $ATT(g,t)$ with clearly defined control groups (i.e., "not-yet-treated" and "never-treated" groups) and then using the probability of belonging to a group to aggregate the $ATT(g,t)$. Therefore, the "negative weights problem" is avoided by using a transparent weighting process and the use of well-defined control groups.

3. Code-Implementation of the Methodology

3.1 Calculating the Aggregated Average Treatment Effects

In the following, the two-step procedure of CS (2021) is implemented in form of two separate functions. In the first function `calculating_agg_att`, the partially and overall aggregated average treatment effects (Agg. ATE) are calculated by first calculating the group-time average treatment effects $ATT(g,t)$ and than using the aggregation methodology of CS (2021). The second function `calculate_att_se`, calculates the standard error of each aggregated average treatment effect with multiplier bootstrapping.

The function `calculating_agg_att` is the main function of this term paper and needs several arguments in order to calculate Agg. ATE. The basic structure of the function is inspired by the code application of CS (2021). The core argument is the method-argument, where the user can choose between the simple-weighted ATE, group-specific ATE, calendar time effects, and event-study effects with balanced and unbalanced event-time. The balanced-argument is needed for the case of performing the balanced event study and requires a number indicating for how many periods a group should have already been treated. Moreover, the unconditional_ind-argument decides on whether to use the unconditional parallel trend assumption (PTA) or the conditional PTA, the later one is the default option. If the conditional PTA is used, pre-treatment covariates should be given to the formula-argument. Additionally, the function requires a balanced panel data set and the name of the time indicator variable, the group-indicator variable, the outcome variable, the individual identification variable and the treatment-indicator variable.

```r
# This function needs a post-treatment outcome vector, a pre-treatment outcome
# vector and a treatment vector, while each row is equal to one observation.
unconditional_att <- function(outcome_post, outcome_pre, treatment) {
# The unconditional_att-function calculates the unconditional ATT in the
# calculating_agg_att-function by first estimating the treatment effect for the
# pre- and post-treatment outcome separately.
  results_post <- lm(outcome_post ~ treatment)
  results_pre  <- lm(outcome_pre ~ treatment)
  coef_post <- coef(results_post)[2]
  coef_pre  <- coef(results_pre)[2]
  # Then, the difference between the two treatment effects is taken, which
  # yields the ATT.
  att <- coef_post - coef_pre

  # Saving the results in list
  results <- list()
  results <- list(ATT = att)
}

calculating_agg_att <- function(data,
                                year_input,
                                group_input,
                                outcome_input,
                                id_input,
                                treatment,
                                formula,
                                unconditional_ind = FALSE,
                                method,
                                balanced = NULL
                                ) {

# 1. Standardize variables & basic variables
# In this section, the arguments given to the function are re-defined in order
# to allow a universal usage of the function.
data_original <- data
data_original <- data_original |>
  rename(year = !!sym(year_input),
         group = !!sym(group_input),
         outcome = !!sym(outcome_input),
         id = !!sym(id_input)
         )
unconditional <- unconditional_ind
covariate_formula <- formula

# Determining the unique time periods, groups and the number of unique values
# of the id-variable . Thereby, the timelist is a tx1 vector and grouplist
# a gx1 vector.
timelist <- unique(data_original$year)
grouplist <- sort(unique(data_original$group))
grouplist <- grouplist[grouplist != 0]
n_unique <- length(unique(data_original$id))

# An empty data frame is setup for saving the ATT(g,t)
num_of_cases <- (length(timelist) - 1) * length(grouplist)
attgt.df <- as.data.frame(matrix(NA, nrow = num_of_cases, ncol = 3))
colnames(attgt.df) <- c("attgt", "group", "year")
```

```r
# The number-variable is needed for the Loop, which calculates the different
# ATT(g,t) and is needed for saving the ATT(g,t) in the right row of the
# data frame.
number <- 1

# 2. The double loop calculates the ATT(g,t), i.e., the disaggregated ATE.
# The first loop iterates over all different groups and the inner
# loop iterates over the available periods. Hence, the inner loop calculates
# all the ATT(g,t) of one group.
for (g in grouplist) {
 for (t in timelist) {

 # The data needs to be assigned for every iteration newly, in order to have
 # the full dataset available as later in the code, the data gets subset.
 data <- data_original

 # The ATT(g,t) is calculated for two periods of a treated group g and
 # "never-treated" group. Therefore, (A) a pre-treatment period (or reference year)
 # needs to be determinant and (B) the data needs to include two years with a
 # treatment and control group.

 # (A) The choice of reference year depends on whether the period of the current
 # loop is laying in the pre- or post-treatment period. If the current period
 # of the iteration is laying in the post-treatment period, then the reference year
 # is equal to the first pre-treatment period of a group. Otherwise, the
 # reference year is set equal to the current period of the iteration,
 # which allows to estimate pseudo-ATT(g,t) like the pre-treatment
 # estimators in the dynamic TWFE regression.
 if (t >= g) {
   reference_year <- g - 1
 } else {
   reference_year <- t
 }

 # (B) After having determined the reference year, the dataset can be subsetted
 # given the reference year and the period after the current loop-period.
 data <- subset(data, year %in% c(reference_year, t + 1))

 # (C) The two-year dataset gets further restricted by only including the treatment
 # group of the current iteration and the "never-treated" group. Thus, a
 # index-indicator is needed to get a dataset, which only includes these two groups.
 data$g_ <- ifelse(data$group == g, 1, 0)
 data$c_ <- ifelse(data$group == 0, 1, 0)
 index_gc <- (data$g_ == 1) | (data$c_ == 1)
 # The actual subsetting of the restricted data.
 data_sel <- data[index_gc,]

 # Create treatment indicator based on the current group in the outer-loop.
 data_sel <- data_sel |> mutate(treat = ifelse(group == g, 1, 0))

 # This if-condition makes sure that the subsetted dataset includes exactely
 # two periods and jumps into the next loop if it is not the case.
 if (length(unique(data$year)) == 2) {
```

```
# (D) The functions for calculating the conditional and unconditional ATT(g,t)'s,
# need data in wide format in which the outcome variable is separately available as
# outcome in the pre- and post-treatment period. Thus, rows of the wide dataset
# is equal to the number of unique observations/id-variable (The code is
# an implementation of BMisc::panel2cs2. see Callaway (2024)). The change
# to the wide format allows later to have clustered standard errors.
data_wide <- arrange(data_sel, id, year)
data_wide$out_post <- data_wide$outcome
data_wide$out_pre <- shift(data_wide$outcome, 1)
data_wide$index <- ifelse(data_wide$year == reference_year, 1, 0)
data_wide <- data_wide[data_wide$index == 0,]
} else {
  next
}

# (F) The final step is to compute the conditional and unconditional ATT(g,t)
# for group at period t. Thereby, one can simply choose between these two
# options by simply setting the argument "unconditional" respectively to the
# preferred estimator.

# Creating a matrix with all values of the pre-treatment covariates
covariates <- model.matrix(covariate_formula, data_wide)

if (unconditional == TRUE) {
# The first option is to calculate the ATT(g,t) with the PTA holding unconditionally.
# The estimation has already been explained in detail in the beginning of this
# section and is a classical canonical DiD ATE.
att <- unconditional_att(outcome_post = data_wide$out_post,
                         outcome_pre  = data_wide$out_pre,
                         treatment    = data_wide$treat)

} else {
# The second option is to calculate the ATT(g,t) with the PTA holding conditional
# to pre-treatment covariates following the non-parametric identified doubly-robust (DR)
# DiD estimation of Sant’Anna & Zhao (2020). The DR DiD combines the outcome regression
# estimator (OR) and inverse probability weighting estimator (IPW).
# Thereby, Sant'Anna & Zhao (2022) stress out that the OR relies on a
# correctly modeled outcome evolution of the control group conditional on
# covariates, while the IPW only relies on the correctly modeled probability
# of belonging to group g. The DR DiD estimator is a combination of the OR
# and IPW and has the advantage that it yields an identified estimator,
# if either the outcome evolution or the probabilities of belonging to a
# group is correctly estimated. This is implemented in the DRDID-package
# of Sant’Anna & Zhao (2020). Thereby, y1 is the required post-treatment outcome vector,
# y0 is the pre-treatment outcome vector and D the treatment vector.
att <- DRDID::drdid_panel(y1 = data_wide$out_post,
                          y0 = data_wide$out_pre,
                          D  = data_wide$treat,
                          covariates = covariates,
                          inffunc    = FALSE,
                          boot       = FALSE)
}
```

```
 # All ATT(g,t) effects are saved in this data frame and is the final
 # result of this loop.
 attgt.df[number, 1] <- att$ATT
 attgt.df[number, 2] <- g
 attgt.df[number, 3] <- t + 1

 # Number is increased after every iteration by one in order to save the
 # values of the next iteration in the next row of the data frame.
 number <- number + 1
 }
}

# 3. Aggregated ATT(g,t) according to CS (2021) ----

 # Set up a copy of the original dataset as the original dataset is later needed
 # for the event study effects.
 data <- data_original

 # After having determined the ATT(g,t) for each group in each period, the
 # weights for aggregating the different ATT(g,t) need to be calculated.
 # These weights are equal to the probability of belonging to a treated group
 # and are calculated in the following..
 # a) An empty dataset for saving the weights is created.
 weights <- as.data.frame(matrix(NA, nrow = length(grouplist), ncol = 3))
 colnames(weights) <- c("group", "size", "probs")

 # b) The weights of each group are calculated in this loop. seq_along
 # creates a number of vectors with length of grouplist, on which the loop iterates on.
 for (i in seq_along(grouplist)) {
  # Selecting the group for which the weight is calculated with the vector of unique groups
  g <- grouplist[i]

  weights[i, 1] <- g
  # The weights are determined by calculating the size of the subgroup g
  weights[i, 2] <- nrow(data[data$group == g & data$year == g, ])
  # and dividing it by the total size of unique observations in the panel dataset.
  weights[i, 3] <- weights[i, 2] / n_unique
 }

 # c) The probabilities of each group are getting merged with the ATT(g,t) by groups.
 # This dataset is playing a key role for the calculation of the Agg. ATE.
 attgt_probs_df <- merge(attgt.df, weights, by.x = "group" )

 # The index_post variable is a index, which contains the row of each ATT(g,t)
 # for each group in the post-treatment period.
 index_post   <- which(attgt_probs_df$year >= attgt_probs_df$group)

## 3.1 Simple Weighted Average of ATT(g,t) ----
if (method == "simple_att") {
 # The simple-weighted average of ATT(g,t) is a measure to evaluate the overall
 # ATE of the treated. Therefore, all the ATT(g,t) of all groups in the
 # post-treatment period are weighted with the respective group weights. Then,
 # all the weighted ATT(g,t) are summed up and this sum is than divided by kappa.
 # This line selects all the relevant ATT(g,t) in the post-treatment period with
 # index_post.
 aggte_simple <- attgt_probs_df[index_post, ]

 # Kappa has the purpose to normalize the weights and is the sum of probabilities
 # corresponding to ATT(g,t) in the post-treatment period over all groups.
 kappa <- sum(aggte_simple$probs)

 # The simple-weighted ATE is calculated according to the description above.
 simple_att_est <- round(sum(aggte_simple$attgt * aggte_simple$probs) / kappa, 4)

 # This line saves the results.
 results <- list(overall_att = simple_att_est)
 return(results)
 }

## 3.2 Group-Time ATT(g,t) ----
if (method == "group_specific_att") {
 # 1) Partially Agg. ATE
 # The group-specific ATE is relevant for researchers if they want to
 # evaluate the heterogeneity of a policy with respect to treatment timing.

 # The first step is to select the ATT(g,t) of each group in the post-treatment
 # period from the data frame with all ATT(g,t).
 gte_df <- attgt_probs_df[index_post, ]

 # An empty data frame is created, in which the group-specific ATE are saved.
 gte_results <- data.frame(matrix(NA, nrow = length(grouplist), ncol = 2))
 colnames(gte_results) <- c("time", "coef")

 # The second step is to calculate the group-specific ATE by taking the
 # mean of all ATT(g,t) of a group g according to Sant'Anna (2022). In each
 # loop iteration, the group-specific ATE is calculated for another group.
 for (i in seq_along(grouplist)) {
  g <- grouplist[i]
  gte_results[i, "time"] <- g
  gte_results[i, "coef" ]  <- round(mean(gte_df[gte_df$group == g, "attgt"]), 4)
 }

 # 2) Overall Agg. ATE
 # The overall ATE of the group-specific ATE is calculated by taking the sum of
 # the weighted partially Agg. ATE. This sum is divided by sum of all group
 # weights in order to normalize the overall Agg. ATE. Thus, the calculation
 # follows the same logic as the one of simple-weighted ATE.
 agg_gte_results <- round(sum(gte_results$coef * weights$probs) / sum(weights$probs), 4)

 # This line saves the results.
 results <- list(partial_att = gte_results, overall_att = agg_gte_results)
 return(results)
}

## 3.3 Calendar Time Effects ----
if (method == "calendar_att") {
```

```r
 # 1) Partially Agg. ATE
 # Calendar time effects are of relevance if the question arises on how large
 # the ATE after a certain number of periods is (CS, 2021). The calendar time
 # effects are calculated by taking the sum of all ATT(g,t) of one year with the
 # corresponding weights of each group and divide by the sum of the weights used
 # in the this year.

 # Selecting the ATT(g,t) of each group in the post-treatment.
 aggte_ct <- attgt_probs_df[index_post,]

 # The calendar time effects are calculated by looping over all unique post-treatment
 # periods and, hence, these unique periods need to be determined.
 calendar_timelist <- unique(aggte_ct$year)

 # Than, an empty data frame is created, in which the calendar time effects are save.
 cte_results <- as.data.frame(matrix(NA, nrow = length(calendar_timelist), ncol = 2))
 colnames(cte_results) <- c("time", "coef")

 # In this loop, the calendar effects for each post-treatment period is calculated.
 for (i in seq_along(calendar_timelist)) {
  # First, the ATT(g,t) of the current period-iteration is selected.
  calendar_time <- calendar_timelist[i]
  df <- aggte_ct[aggte_ct$year == calendar_time,]
  # Secondly, the calendar time effects are calculated by weighting the ATT(g,t) with the
  # corresponding weights in one year and divide it by the sum of all used weights
  # in this period.
  cte_results[i, "time"] <- calendar_time
  cte_results[i, "coef"] <- round(sum(df$attgt * df$probs) / sum(df$probs), 4)
 }

 # 2) Overall Agg. ATE
 # The overall calendar time effects is calculated by taking the mean of all
 # calendar time effects.
 agg_att_ct <- round(mean(cte_results$coef), 4)

 # Save final calendar time effects
 results <- list(partial_att = cte_results, overall_att = agg_att_ct)
 return(results)
}

## 3.4 Event Study Design ----
if (method %in% c("unbalanced_eventstudy", "balanced_eventstudy")) {
 # In this section the event study effects is calculated for the balanced and
 # unbalanced groups. The interest lays on the heterogeneity of ATE with respect
 # to exposure to the treatment. In the first step, the event-time needs to be
 # calculated by taking the difference between the years and groups.
 attgt_et <- attgt_probs_df
 attgt_et$eventtime <- attgt_et$year - attgt_et$group

 # Than, all pre-treatment event times are excluded as the interest lays on the
 # exposure to the treatment period.
 attgt_et <- attgt_et[attgt_et$eventtime >= 0, ]
```

```r
### 3.4.1 Event Study Effect ----
if (method == "unbalanced_eventstudy") {
 # 1) Partial Agg. ATE
 # The event study effects without balancing the event time measures the
 # effect of a treatment e periods after a treatment. Therefore, the weighted
 # sum of all ATT(g,t) of an event-time e is taken and divided by the
 # sum of weights of all groups in event-time e.

 # The calculation of the event study effect requires a variable with the
 # unique set of event-times. Than, the loop can be used to calculate the
 # event study effects for each event-time e.
 eventime_timelist <- unique(attgt_et$eventtime)

 # Creating an empty dataset to safe the event study effects of each event-time e
 et_results <- as.data.frame(matrix(NA, nrow = length(eventime_timelist), ncol = 2))
 colnames(et_results) <- c("time", "coef")

 # This loop calculates the event-study effects for each even-time.
 for (i in seq_along(eventime_timelist)) {
  # Selecting the event-time e of the current iteration
  eventime_time <- eventime_timelist[i]
  df <- attgt_et[attgt_et$eventtime == eventime_time,]
  # Calculating the event study effect according to the aforementioned description.
  et_results[i, "time"] <- eventime_time
  et_results[i, "coef"] <- round(sum(df$attgt * df$probs) / sum(df$probs), 4)
 }

 # 2) Overall Agg. ATE
 # The overall event study effect is calculated by taking the mean of all
 # event study effects time effects.
 agg_et <- round(mean(et_results$coef), 4)

 # Save final event study effects
 results <- list(partial_att = et_results, overall_att = agg_et)
 return(results)
}

## 3.4.2 Eventstudy with balanced group ----
if (method == "balanced_eventstudy" & !is.null(balanced)) {

 # 1) Partially Agg. ATE
 # The event study effect with groups being balanced with respect to the
 # event-time e condition the event study effect on a group being observed
 # at least for a certain amount of time according to CS (2021). Thereby,
 # the calculation of the "balanced event study effect" is identical to the
 # "unbalanced event study effect".

 # (A) First, the indicator on how many periods a group should have been
 # observed is re-defined as it is an argument of the function.
 balance_groups <- balanced

 # (B) Secondly, all the groups, which were treated at least for the certain
 # amount of periods, must be determined. Thus, the number of post-treatment
 # periods is calculated by iterating over the vector with the unique treatment groups.
 grouplist_et <- unique(attgt_et$group)
 att_per_group <- sapply(grouplist_et, function(g){
  # Create a subset of the ATT(g,t) with only group g being represented.
  df <- attgt_et[attgt_et$group == g,]
  # Than the number of post-treatment period is determined.
  n  <- nrow(df)
  n
 })

 # Create a data frame, which contains the information on the number
 # of post-treatment periods.
 att_per_group <- data.frame(grouplist_et, att_per_group)
 colnames(att_per_group) <- c("eventtime", "abs_nr_att")

 # The loop determines whether a group was exposed to the treatment for
 # the required periods determined by "balanced_group".
 groups_to_exclude <- c()
 for (i in seq_along(att_per_group$eventtime)) {
  # The groups with "balance_group + 1" period are excluded in the if-condition.
  if (att_per_group[i, 2] < balance_groups + 1) {
  #"groups_to_exclude" is containing the groups, which were not observed for required treatment length.
   groups_to_exclude[i] <- att_per_group[i, 1]
   } else {
     next
   }
  }

 # In the last step, the "balanced" data frame of ATT(g,t) is obtained by
 # excluding the determined groups in "groups_to_exclude".
 groups_to_exclude <- groups_to_exclude[!is.na(groups_to_exclude)]
 attgt_et_bal <- attgt_et[!attgt_et$group %in% groups_to_exclude,]

 # (C) Now, the event study effect can be calculated in the same way as the
 # in the unbalanced event study effect section.
 eventime_timelist <- unique(attgt_et_bal$eventtime)
 att_et_bal <- as.data.frame(matrix(NA, nrow = length(eventime_timelist), ncol = 2))
 colnames(att_et_bal) <- c("time", "coef")

 for (i in seq_along(eventime_timelist)) {
  eventime_time <- eventime_timelist[i]
  df <- attgt_et[attgt_et_bal$eventtime == eventime_time,]
  att_et_bal[i, "time"]   <- eventime_time
  att_et_bal[i, "coef"]   <- round(sum(df$attgt * df$probs) / sum(df$probs), 4)
 }

 # Only these event study effects are shown, which lay in the minimum requirement
 # of how long a group must have been observed.
 et_bal_results <- att_et_bal[att_et_bal$time < balance_groups + 1,]

 # 2) Overall Agg. ATE
 # Calculate the aggregated event study effects by taking the mean of all
 # event study effects in the different event-time.
```

```
 agg_bal_et <- round(mean(et_bal_results$coef), 4)

 # Save results on balanced event study effects
 results <- list(partial_att = et_bal_results, overall_att = agg_bal_et)
 return(results)
}
}
}
```

3.2 Calculation of the Standard Errors

The function `calculate_att_se` is the final step of implementing the methodology of CS (2021). The outcome of the function is a list of the partially and overall Agg. ATE of a chosen method and the corresponding standard errors. The clustered standard errors are calculated with the multiplier bootstrap procedure proposed in CS (2021). Thereby, it is important to mention that the multiplier bootstrap in CS (2021) is used to bootstrap the influence function of an ATE, while in this code it is implemented to calculate a distribution for each partial and overall ATT. This allows then to calculate the standard errors, which are clustered on unit-level, for each partial and overall ATT based on this distribution of ATT values.

```
calculate_att_se <- function(data = qwi,
                             year_input = "date_y",
                             group_input = "group",
                             outcome_input = "lnEmp",
                             id_input = "county_id",
                             treatment = treated,
                             formula = spec_formula,
                             unconditional_ind = FALSE,
                             method = "group_att",
                             balanced = FALSE) {

 # The data is re-defined and the id_input variable name is changed to "id"
 data_for_b <- data |>
  rename(id = !!sym(id_input)) |>
  as.data.frame()

 # The number of bootstrap-iteration is determined here. The larger the number
 # the more the standard errors converge to their population equivalent.
 iter <- 100

 # Creating the list for the distribution of the partial and overall ATT
 b_res_overall <- list()
 b_res_partial <- list()

 # This loop is the implementation of the multiplier bootstrap. The core
 # difference to the standard bootstrapping is that there is no re-sampling of
 # observations. Instead only a fraction of the overall total sample size is
 # used for calculating the different ATE.
 for (i in 1:iter) {

  # The selection of a subset of the total sample is determined by the
  # Bernoulli Variates following the proposed Algorithm of CS (2021). Thereby,
  # a binomial distribution is created for each iteration and is saved in
  # vector "bernoulli_variates". The vector has the same length as the unique
  # observation of the dataset. The probability used for the binomial
 # distribution is the following term of CS (2021):
 kappa <- (sqrt(5) + 1) / 2
 p <- kappa / sqrt(5)
 n_row <- length(data_for_b$id)
 bernoulli_variates <- rbinom(n_row, 1, p)

 # In these three lines the subset of the total sample is drawn based on the
 # Bernoulli Variates. Additionally, it is to mention that the wide format
 # of the data allows to estimate cluster standard errors on the unit-level.
 county <- unique(data_for_b$id)
 county <- county[bernoulli_variates == 1]
 b_data <- data_for_b |> filter(id %in% county)

 # This subset is then used to calculate the partial and overall ATE of the
 # method of choice for a loop iteration.
 b_est <- calculating_agg_att(data = b_data, year_input = year_input,
                              group_input = group_input, outcome_input = outcome_input,
                              id_input = "id", treatment = treated, formula = formula,
                              unconditional_ind = unconditional_ind, method = method,
                              balanced = balanced)

 # The results of the ATE in the current loop-iteration are saved here, and these
 # list contain the distribution of every partial and overall ATE after
 # the loop has come to its end.
 if (method != "simple_att")  b_res_partial[[i]] <- t(as.matrix(b_est$partial_att$coef))
  b_res_overall[[i]] <- round(b_est$overall_att, 4)
}

# First, the lists, which contains the ATT must be collapsed and are converted
# into a data frame.
b_res_partial <- map_dfr(b_res_partial, as.data.frame)
b_res_overall <- unlist(b_res_overall)

# Then, the standard error of the partial and overall Agg. ATE are calculated
# from their respective distribution of ATE.
se_partial <- apply(b_res_partial, 2, sd)
se_overall <- round(sd(b_res_overall), 4)

# After calculating the standard errors, the partial and overall Agg. ATE are
# calculated for the full sample.
est <- calculating_agg_att(data = data, year_input = year_input, group_input = group_input,
                           outcome_input = outcome_input, id_input = id_input,
                           treatment = treated, formula = spec_formula,
                           unconditional_ind = unconditional_ind, method = method,
                           balanced = balanced)

# The results of the partial and overall Agg. ATE and the corresponding
# standard errors are saved as the outcome of this function.
est$partial_att$b_se <- round(se_partial, 4)
est$overall_att <- cbind(est$overall_att, se_overall)
colnames(est$overall_att) <- c("coef", "b_se")

return(est)
```

```
}
```

4. Application of Callaway & Sant'Anna (2021)

Callaway & Sant'Anna (2021) explore the effect of an increase in the minimum wage on teen employment in the United States. Therefore, the authors take advantage of the situation during the period between 2001 and 2007, where several states increased their minimum wage above the federal minimum wage in different years, while the federal minimum wage stayed constant. This allows CS (2021) to exploit the variation on state-level with respect to minimum wage increases and enables them to calculate ATE.

In this term paper, the same application of CS (2021) is performed with the code-implementation presented above. Thereby, the dataset used by CS (2021) is recreated by using the information on teen employment from the public available Quarterly Workforce Indicator (QWI) on county-level and information on the pre-treatment characteristics of counties in the U.S. from the 2000 County and City Databook. The last dataset allows to control for several county-specific characteristics such as county population in 2000, fraction of the white population, share of the population with high-school diploma, the median income in 1997 and share of population below the poverty level. These values are used in their logarithmized-version for controlling the county-specific trends and allow to implement the conditional PTA like in CS (2021).

The effect of the minimum wage on teen employment is evaluated on the full sample as in CS (2021), which contains three treated groups, i.e., groups 2004, 2006 and 2007. Moreover, the research is extended to evaluate the effect on Census-Region-level defined by the U.S. Census Bureau. The following regions are observed: the Western, Southern and Midwest region. Thereby, all three groups have a different number of treatment groups. While the Midwestern region has all three groups, the Southern region only contains groups 2006 and 2007 and the Western group only has group 2007.

In the following, the results are discussed after the table structure is explained. In "twfe", the TWFE estimator can be observed and in "simple", the simple-weighted average Agg. ATT can be found with the bootstrapped standard error. The table structure has a simple structure from here on, the partial effect can be found under partial_effects and the overall Agg. ATE in "overall_att". In "group", the group-specific effects can be found and the time represents the treatment group, e.g., row one represent the group-specific effect of group 2004. In "calendar", the calendar time effects can be found, while the column "time" represents a year. In "eventstudy_bal" & "eventstudy_unbal", the event study effects for each event-time e can be found.

In table (1), the TWFE effect has a positive but insignificant estimator. A completely different picture can be seen if the simple-weighted average ATT is observed, which suggests a 4.45% decrease in teen employment, due to the minimum wage increase. The group-specific ATT shows that group 2004, the one with the longest post-treatment period, also has the greatest ATT as teen employment decreases by 8.94%. The group-specific effect declines with a shortening post-treatment period. The calendar time effects can be interpreted as the ATE of all groups in a period t. Thus, in row 2007, the interpretation would be that teen employment decreases by 4.03% across all groups in 2007. It is important to point out that the calendar time effects are stable over the years. In the balanced event-study, only groups are included, which were observed for more than one period, i.e., group 2007 is excluded. The effect seems to grow with longer exposure to treatment, which can be seen by the increasing magnitude of the negative effect over time for both event study effects.

```
## [1] "(1) Full sample: Conditional ATE with Doubly-Robust estimators"
```

```
## $twfe
##     coef     se
## 1 0.0048 0.0069
##
## $simple
## $simple$overall_att
##         coef   b_se
## [1,] -0.0445 0.0045
```

```
##
## $simple$partial_att
## $simple$partial_att$b_se
## numeric(0)
##
##
##
## $group
## $group$partial_att
##   time    coef   b_se
## 1 2004 -0.0894 0.0131
## 2 2006 -0.0404 0.0053
## 3 2007 -0.0167 0.0043
##
## $group$overall_att
##         coef   b_se
## [1,] -0.0306 0.0034
##
##
## $calendar
## $calendar$partial_att
##   time    coef   b_se
## 1 2004 -0.0497 0.0140
## 2 2005 -0.0646 0.0131
## 3 2006 -0.0485 0.0053
## 4 2007 -0.0403 0.0038
##
## $calendar$overall_att
##         coef   b_se
## [1,] -0.0508 0.0076
##
##
## $eventstudy_bal
## $eventstudy_bal$partial_att
##   time    coef   b_se
## 1    0 -0.0212 0.0034
## 2    1 -0.0619 0.0057
##
## $eventstudy_bal$overall_att
##         coef   b_se
## [1,] -0.0415 0.0041
##
##
## $eventstudy_unbal
## $eventstudy_unbal$partial_att
##   time    coef   b_se
## 1    0 -0.0212 0.0032
## 2    1 -0.0619 0.0061
## 3    2 -0.1107 0.0116
## 4    3 -0.1325 0.0161
##
## $eventstudy_unbal$overall_att
##         coef   b_se
## [1,] -0.0816 0.0078
```

In table (2), the unconditional ATT for the full sample can be observed. Thereby, it is important to recognize that the effect sizes are similar for every treatment effects of table (1) regardless of the method. This should not be the case because one would expect the ATT's to lay higher, due to the uncontrolled country-specific trends. One reason for the difference could be the data itself, as CS (2021) give little detail on how the data was matched and which variables exactly were used. Thus, the collection of the dataset could be the core reason for this result.

```
## [1] "(2) Full sample: Unconditional ATE"
```

```
## $twfe
##     coef     se
## 1 0.0048 0.0069
##
## $simple
## $simple$overall_att
##         coef   b_se
## [1,] -0.0461 0.0045
##
## $simple$partial_att
## $simple$partial_att$b_se
## numeric(0)
##
##
##
## $group
## $group$partial_att
##   time    coef   b_se
## 1 2004 -0.0793 0.0131
## 2 2006 -0.0553 0.0046
## 3 2007 -0.0163 0.0044
##
## $group$overall_att
##         coef   b_se
## [1,] -0.0328 0.0035
##
##
## $calendar
## $calendar$partial_att
##   time    coef   b_se
## 1 2004 -0.0448 0.0140
## 2 2005 -0.0483 0.0131
## 3 2006 -0.0565 0.0051
## 4 2007 -0.0423 0.0042
##
## $calendar$overall_att
##        coef   b_se
## [1,] -0.048 0.0077
##
##
## $eventstudy_bal
## $eventstudy_bal$partial_att
##   time    coef   b_se
## 1    0 -0.0240 0.0033
## 2    1 -0.0673 0.0052
##
```

```
## $eventstudy_bal$overall_att
##         coef   b_se
## [1,] -0.0456 0.0037
##
##
## $eventstudy_unbal
## $eventstudy_unbal$partial_att
##   time    coef   b_se
## 1    0 -0.0240 0.0038
## 2    1 -0.0673 0.0060
## 3    2 -0.1043 0.0113
## 4    3 -0.1199 0.0152
##
## $eventstudy_unbal$overall_att
##         coef   b_se
## [1,] -0.0789 0.0078
```

In the next step, the regional effects of the conditional ATE are discussed on how they differ to the results of the full sample. The discussion on the Western region is excluded because states of this region only increased in the minimum wage in 2007. Therefore, the effects for each aggregation ATE are the same as there is no variation in treatment timing.

In table (3), the results on the Midwest region can be interpreted and a great similarity between the ATE with conditional and unconditional PTA can be observed. The group-specific effects of the Midwest region for the groups 2004 and 2006 are nearly only half the magnitude as in the full sample, while the group 2007 has a double as large estimator. The calendar time effects are comparable to the ones in the full sample. The event study effects are showing a different pattern compared to the full sample. The event study effects for balanced and unbalanced data are increasing in magnitude more slowly in the Midwest region, which could lead to the conclusion that this region may have a smaller problem with teen employment than other regions.

```
## [1] "(3) Midwest Census Region: Conditional ATE"
```

```
## $twfe
##      coef     se
## 1 -0.0065 0.0083
##
## $simple
## $simple$overall_att
##       coef   b_se
## [1,] -0.04 0.0059
##
## $simple$partial_att
## $simple$partial_att$b_se
## numeric(0)
##
##
##
## $group
## $group$partial_att
##   time    coef   b_se
## 1 2004 -0.0549 0.0122
## 2 2006 -0.0338 0.0058
## 3 2007 -0.0254 0.0064
##
## $group$overall_att
##         coef   b_se
## [1,] -0.0334 0.0048
##
##
## $calendar
## $calendar$partial_att
##   time    coef   b_se
## 1 2004 -0.0412 0.0187
## 2 2005 -0.0356 0.0178
## 3 2006 -0.0390 0.0078
## 4 2007 -0.0410 0.0065
##
## $calendar$overall_att
##          coef   b_se
## [1,] -0.0392 0.0112
##
##
## $eventstudy_bal
## $eventstudy_bal$partial_att
##   time    coef   b_se
## 1    0 -0.0279 0.0043
## 2    1 -0.0406 0.0079
##
## $eventstudy_bal$overall_att
##          coef   b_se
## [1,] -0.0343 0.0055
##
##
## $eventstudy_unbal
## $eventstudy_unbal$partial_att
##   time    coef   b_se
## 1    0 -0.0279 0.0050
## 2    1 -0.0406 0.0073
## 3    2 -0.0628 0.0125
## 4    3 -0.0801 0.0176
##
## $eventstudy_unbal$overall_att
##          coef   b_se
## [1,] -0.0528 0.0091
```

In table (4), the mixed results for the Southern region can be interpreted and also there is great similarity between the ATT with conditional and unconditional PTA. The group-specific effects are a lot smaller for this region or are equal to zero as it is the case for group 2007. The calendar time effects show that the average effect in 2006 is insignificantly positive, while the effect is positive in the year 2007 but has only one-fourth of the size of the effects in the full sample. The event study effects for balanced and unbalanced groups in event-time 0 is nearly equal to zero, while a similar effect-magnitude is observed in the full sample in event-time 1.

```
## [1] "(4) Soutern Census Region: Conditional ATE"
```

```
## $twfe
##     coef     se
## 1 0.0773 0.0114
##
## $simple
## $simple$overall_att
##         coef   b_se
## [1,] -0.0082 0.0054
##
## $simple$partial_att
## $simple$partial_att$b_se
## numeric(0)
##
##
##
## $group
## $group$partial_att
##   time    coef   b_se
## 1 2006 -0.0212 0.0090
## 2 2007  0.0013 0.0075
##
## $group$overall_att
##         coef   b_se
## [1,] -0.0047 0.0066
##
##
## $calendar
## $calendar$partial_att
##   time    coef   b_se
## 1 2006  0.0123 0.0090
## 2 2007 -0.0136 0.0068
##
## $calendar$overall_att
##        coef   b_se
## [1,] -6e-04 0.0061
##
##
## $eventstudy_bal
## $eventstudy_bal$partial_att
##   time    coef   b_se
## 1    0  0.0043 0.0059
## 2    1 -0.0548 0.0101
##
## $eventstudy_bal$overall_att
##         coef   b_se
## [1,] -0.0253 0.0069
##
##
## $eventstudy_unbal
## $eventstudy_unbal$partial_att
##   time    coef   b_se
## 1    0  0.0043 0.0062
## 2    1 -0.0548 0.0109
##
## $eventstudy_unbal$overall_att
##         coef   b_se
## [1,] -0.0253 0.0073
```

5. References

Barrett T, Dowle M, Srinivasan A, Gorecki J, Chirico M, Hocking T (2024). data.table: Extension of 'data.frame'. R package version 1.15.99, https://Rdatatable.gitlab.io/data.table, https://github.com/Rdatatable/data.table, https://r-datatable.com.

Callaway, B. (2023). Introduction to DiD with Multiple Time Periods. Retrieved June 2, 2024, from https://bcallaway11.github.io/did/articles/multi-period-did.html

Callaway, B. (2024). BMisc (1.4.6) [Software]. https://github.com/bcallaway11/BMisc

Callaway, B., & Sant'Anna, P. H. (2021). Difference-in-Differences with multiple time periods. Journal of Econometrics, 225(2), 200–230.

Croissant Y, Millo G (2008). "Panel Data Econometrics in R: The plm Package." Journal of Statistical Software, 27(2), 1–43

De Chaisemartin, C., & D'Haultfoeille, X. (2020). Two-Way fixed Effects Estimators with Heterogenous Treatment Effects. American Economic Association, 2964–2996.

Goodman-Bacon, A. (2021). Difference-in-Differences with Variation in Treatment Timing. Journal of Econometrics, 254–277. Sant'Anna, B. C. a. P. H. (2022, July 19). Getting Started with the did Package. https://cran.r-project.org/web/packages/did/vignettes/did-basics.html

U.S. Census Bureau. (2001). County and City Data Book: 2000 [Dataset]. https://www.census.gov/library/publications/2001/compendia/ccdb00.html

U.S. Census Bureau. (2024). Quarterly Workforce Indicator [Dataset]. https://lehd.ces.census.gov/data/

Sant'Anna, P., & Zhao, J. (2020). Doubly Robust Difference-in-Difference Estimators. Journal of Econometrics, 101–122.

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." Journal of Open Source Software, 4(43), 1686.

Zeileis A, Hothorn T (2002). "Diagnostic Checking in Regression Relationships." R News, 2(3), 7–10. https://CRAN.R-project.org/doc/Rnews/.