Extra Analyses

2025-03-14

```
## Thank you for using fastDummies!

## To acknowledge our work, please cite the package:

## Kaplan, J. & Schlegel, B. (2023). fastDummies: Fast Creation of Dummy (Binary) Columns and Rows from

## Loading required package: Matrix

##
## Attaching package: 'readr'

## The following object is masked from 'package:scales':
##
##     col_factor

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x readr::col_factor()  masks scales::col_factor()
## x gridExtra::combine() masks dplyr::combine()
## x purrr::discard()     masks scales::discard()
## x tidyr::expand()      masks Matrix::expand()
## x dplyr::filter()      masks stats::filter()
## x dplyr::lag()         masks stats::lag()
## x tidyr::pack()        masks Matrix::pack()
## x tidyr::unpack()      masks Matrix::unpack()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

1. Load data, preprocess

Implementation for DiD, SC, Synth. DiD, Generate plots for different outcome variables

```
# Get vector of cohort years
cohorts <- sort(unique(dat$treatment_year))
cohorts <- cohorts[!is.infinite(cohorts)]

# Initialize a list to store results
results <- list()

#' Function that implements DiD, SC, and Synth. DiD and plots outcomes
#' @param data panel data with state-year as units
#' @param outcome_var dependent variable we want to measure
#' @param cohorts vector of treatment years present throughout the data
run_synthdid <- function(data, outcome_var, cohorts) {

  outcome_labels <- c(
    pct_in_comp_mco = "Percent in Comp. MCO",
    total_medicaid = "Total Medicaid Spending",
    log_total_spend = "Log(Total Medicaid Spending)",
    all_medicaid_spending_per_cap = "Total Medicaid Spending per Capita",
    log_all_medicaid_spending_per_cap = "Log(Total Medicaid Spending per Capita)"
  )

  outcome_label <- outcome_labels[[outcome_var]]

  # Loop through each cohort
  for (cohort_year in cohorts) {

    tryCatch({
      # Subset data for the current cohort
      cohort_data <- data %>%
        filter(treatment_year == cohort_year | is.infinite(treatment_year))

      # Reshape the cohort data to wide format
      Y <- cohort_data %>%
        select(state, year, !!sym(outcome_var)) %>%
        pivot_wider(names_from = year,
                    values_from = !!sym(outcome_var)) %>%
        column_to_rownames("state") %>%
        as.matrix()

      # Extract treatment timing for the cohort
      treatment_timing <- cohort_data %>%
        group_by(state) %>%
        summarize(treatment_year = unique(treatment_year)) %>%
        pull(treatment_year)

      # Number of control units (NO)
      NO <- sum(treatment_timing == Inf)

      # Number of pre-treatment periods (TO)
```

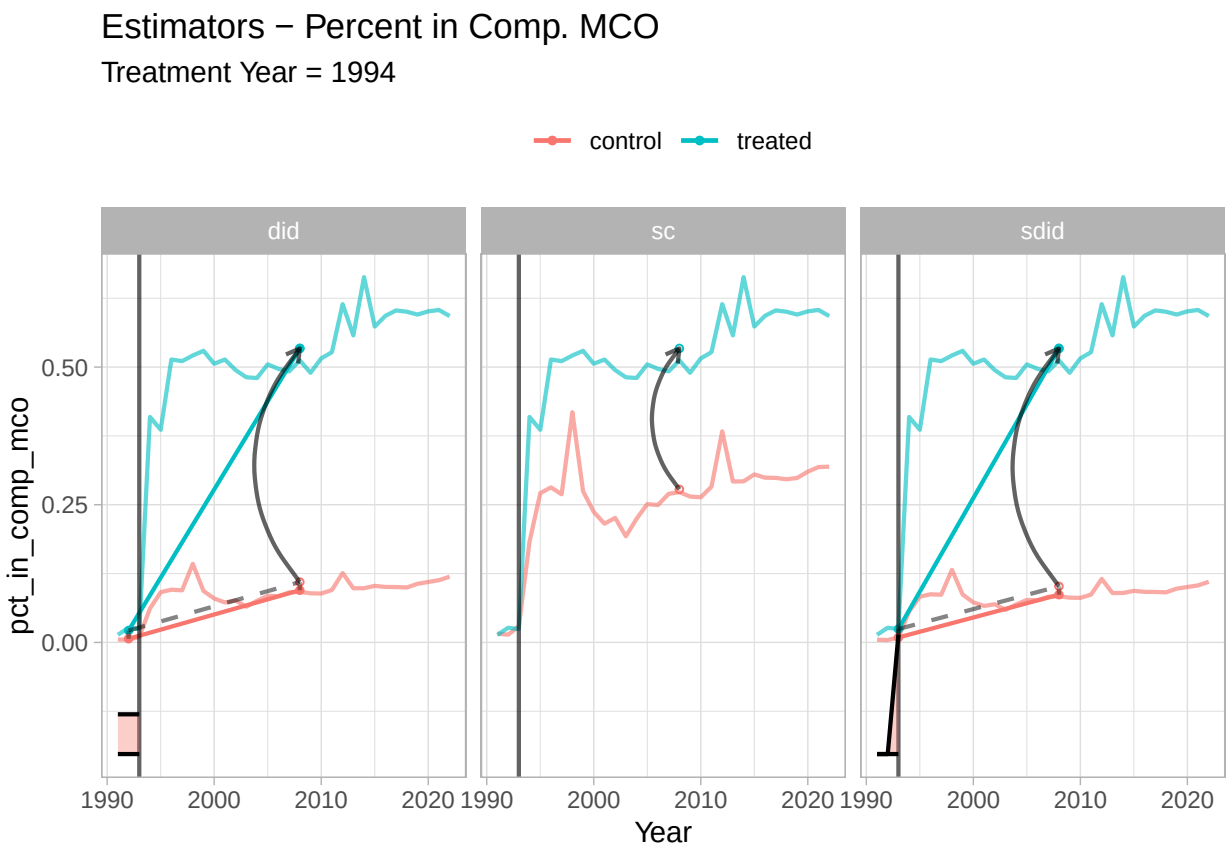
```
      T0 <- cohort_year - min(cohort_data$year)

      # Check if Y has valid dimensions
      if (nrow(Y) == 0 || ncol(Y) == 0) {
        stop("Y has invalid dimensions. Skipping cohort year ", cohort_year)
      }

      # Estimate the treatment effect using synthdid
      tau.hat <- synthdid_estimate(Y, N0, T0)

      # Define estimators
      estimators <- list(did = did_estimate,
                         sc = sc_estimate,
                         sdid = synthdid_estimate)

      # Apply estimators
      estimates <- lapply(estimators, function(estimator) {estimator(Y, N0, T0)})

      # Generate the plot
      plot <- synthdid_plot(estimates[1:3],
                            control.name = 'control',
                            treated.name = "treated",  # Use the treated unit's name
                            facet.vertical = FALSE,
                            lambda.comparable = TRUE,
                            se.method = 'none',
                            trajectory.linetype = 1,
                            line.width = .75,
                            effect.curvature = -.4,
                            trajectory.alpha = .7,
                            effect.alpha = .7,
                            diagram.alpha = 1,
                            onset.alpha = .7) +
        labs(title = paste0("Estimators - ", outcome_label),
             subtitle = paste0("Treatment Year = ", cohort_year),
             x = "Year",
             y = outcome_var)

      print(plot)

      # Print a success message
      message("Plot saved for cohort year ", cohort_year)

    }, error = function(e) {
      # Print an error message and skip the cohort year
      message("Error in cohort year ", cohort_year, ": ", e$message)
    })
  }
}
```

Percent in Comprehensive Managed Care

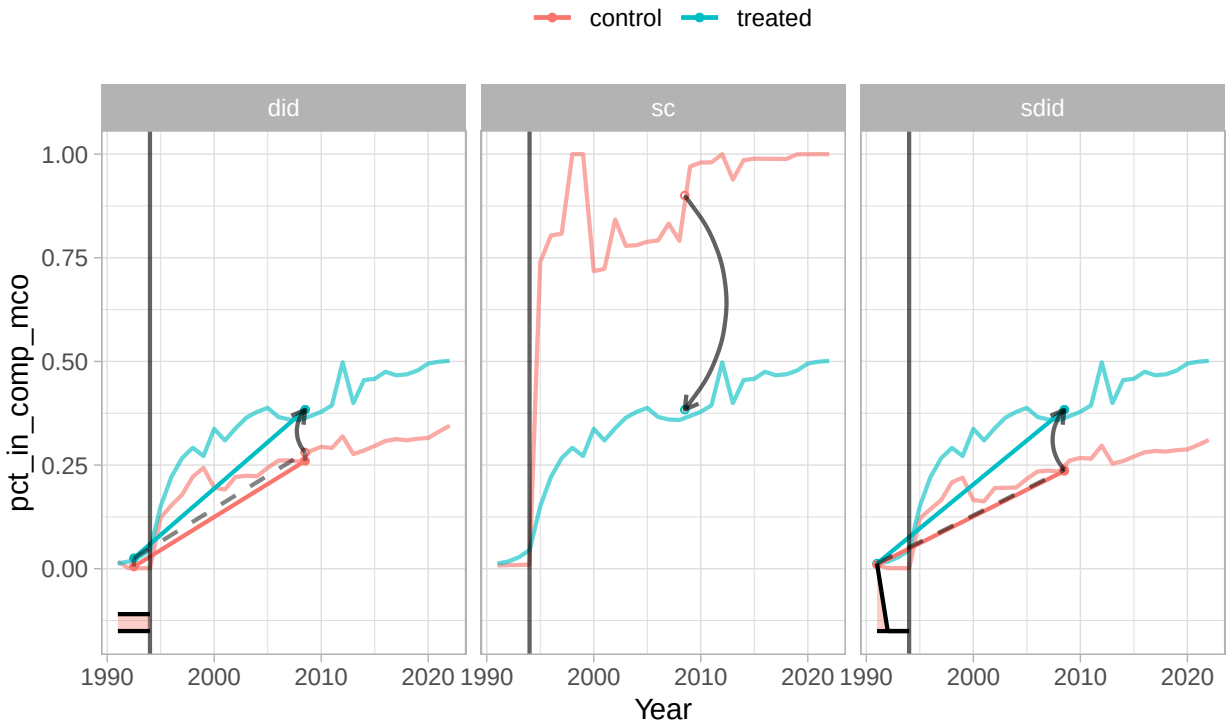
```
run_synthdid(data, "pct_in_comp_mco", cohorts)
```

```
## Error in cohort year 1992: dim(X) must have a positive length
```

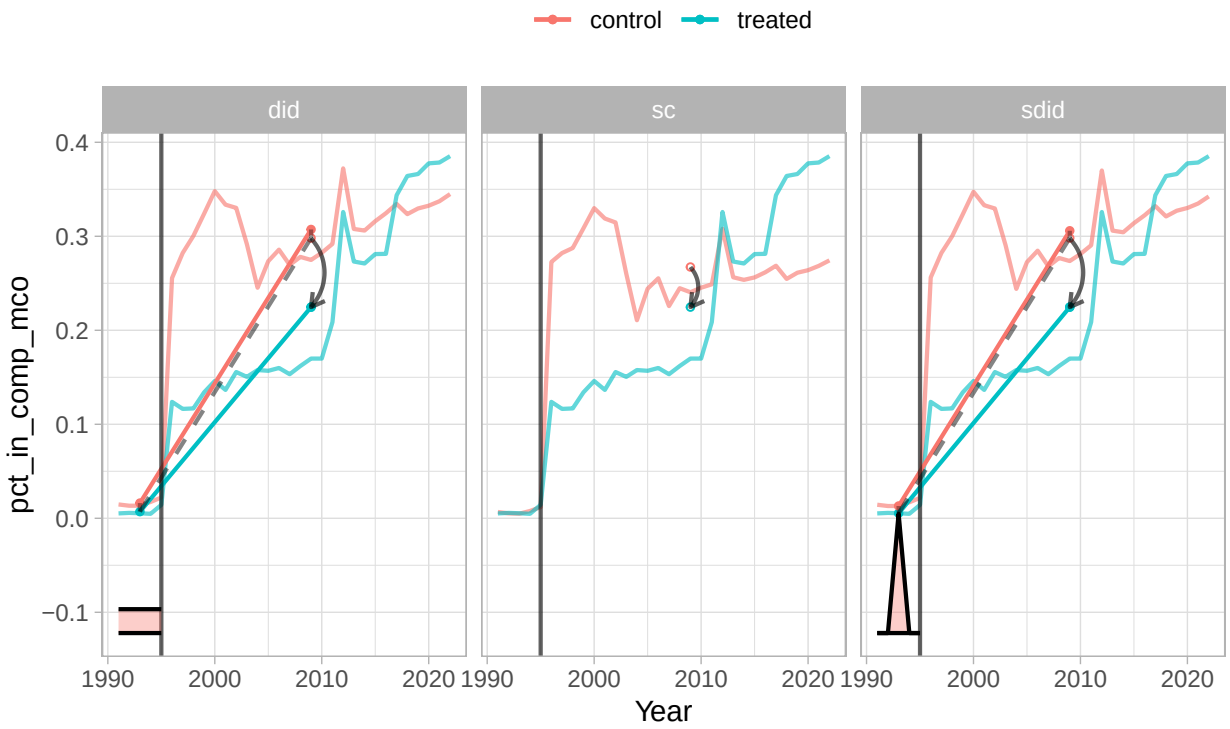
```
## Plot saved for cohort year 1994
```



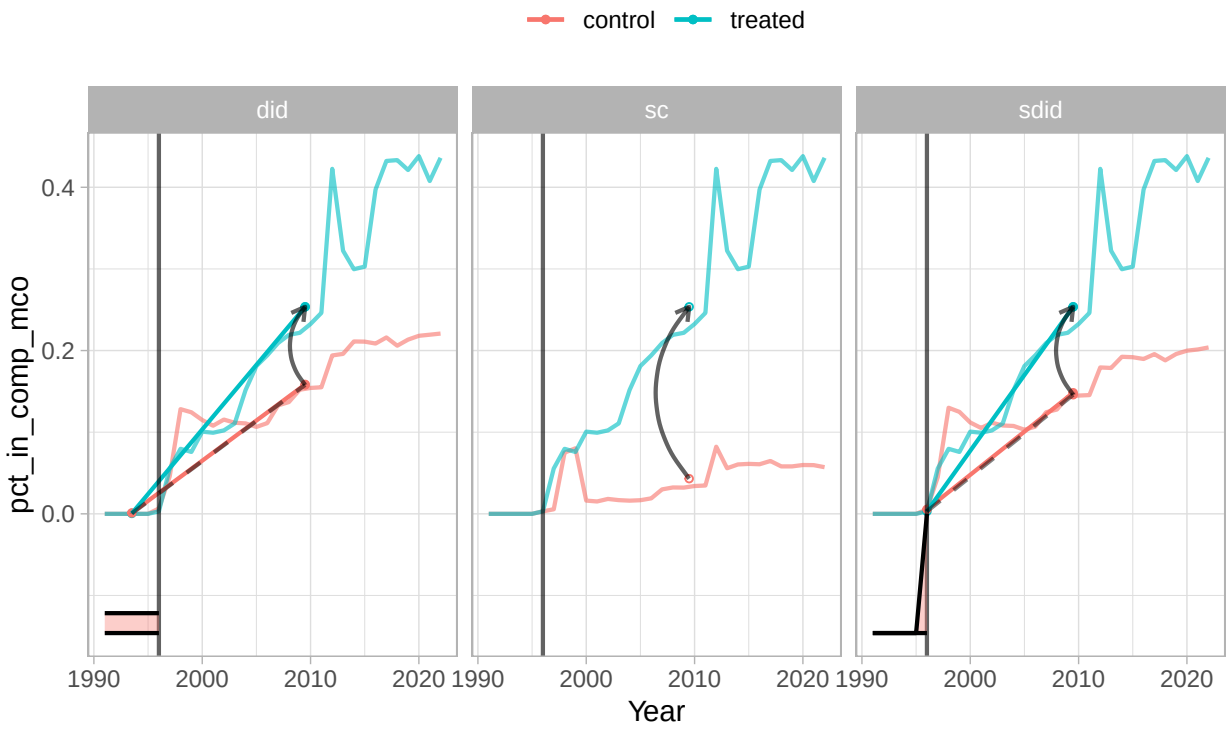

```
## Plot saved for cohort year 1995
```



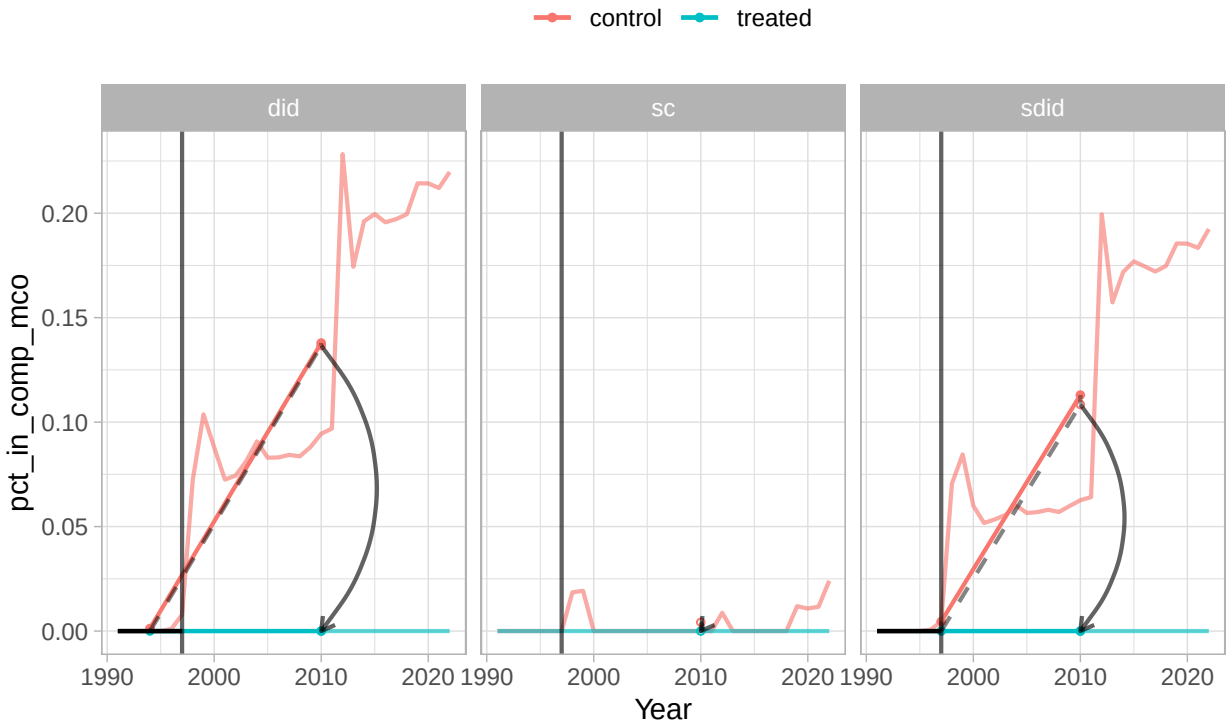

```
## Plot saved for cohort year 1996
```



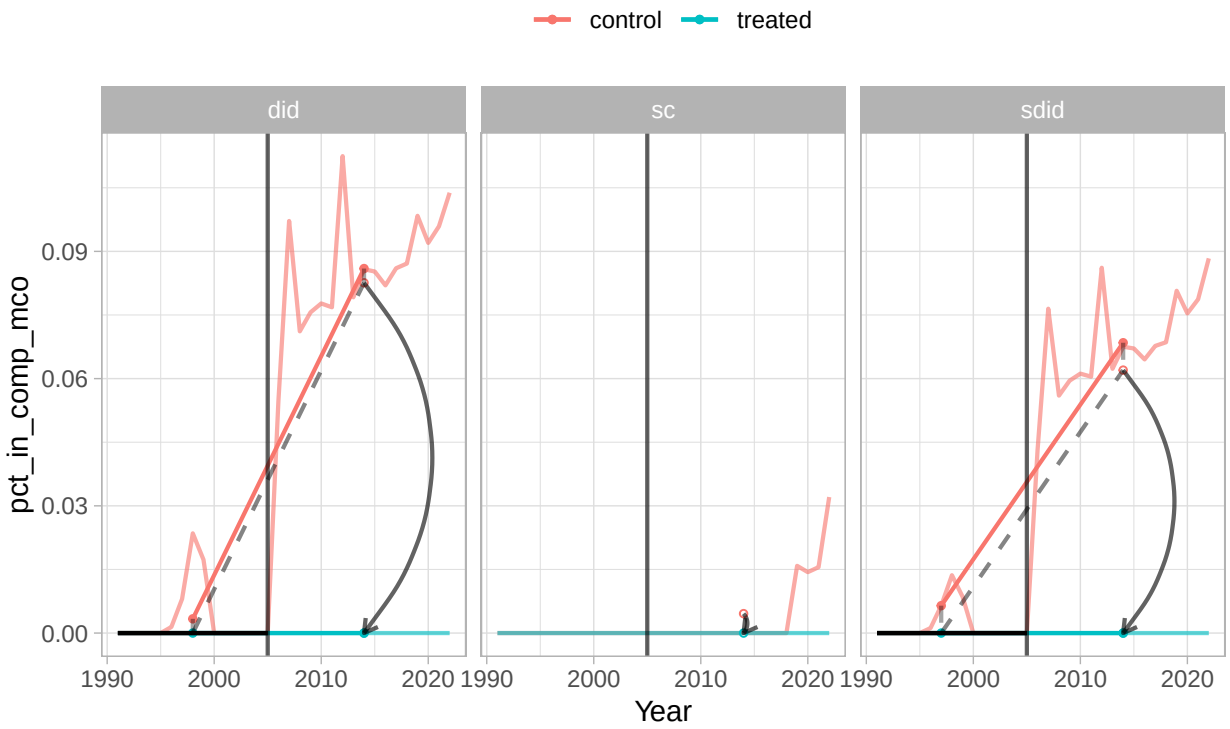

```
## Plot saved for cohort year 1997
```



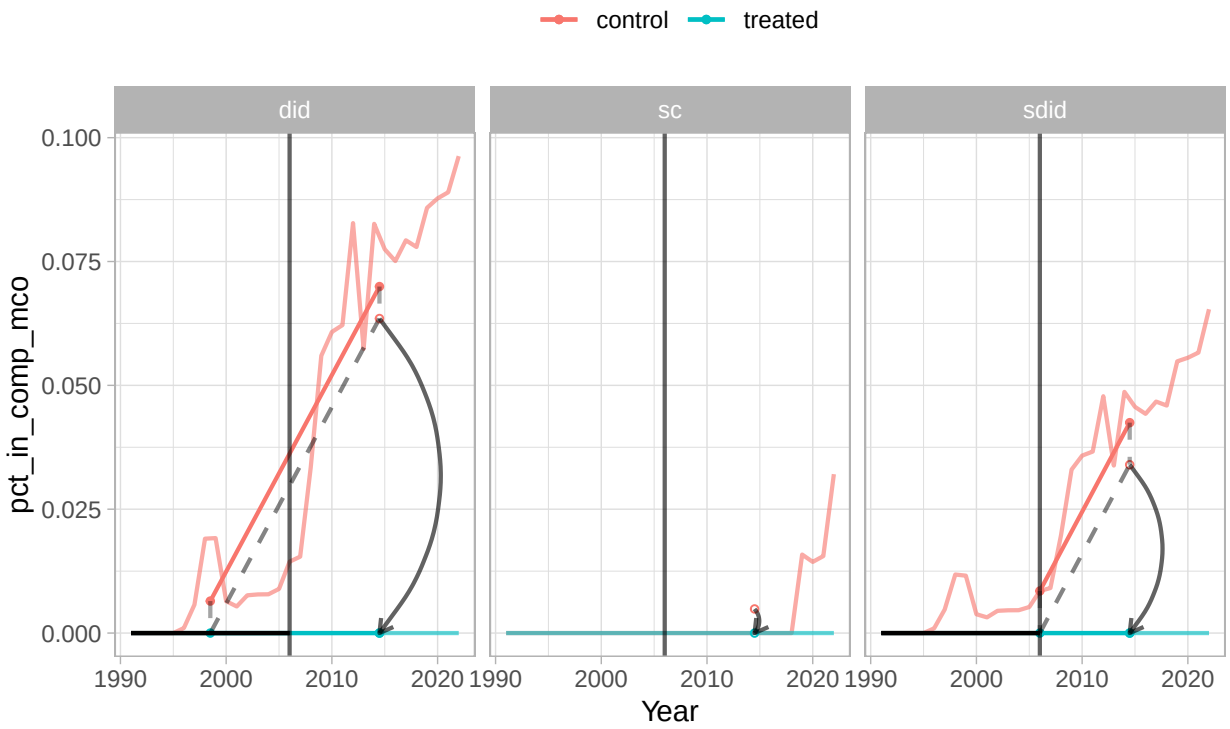

```
## Plot saved for cohort year 1998
```



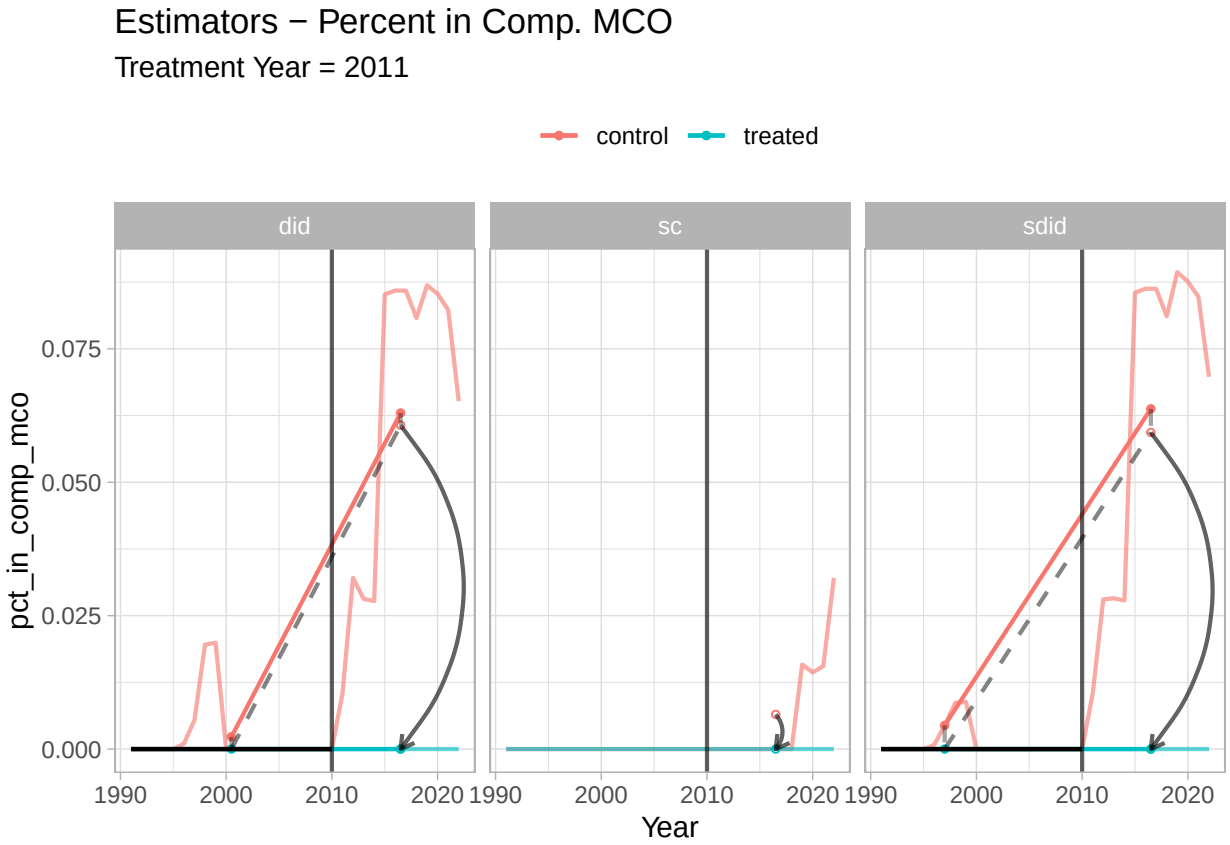

```
## Plot saved for cohort year 2006
```



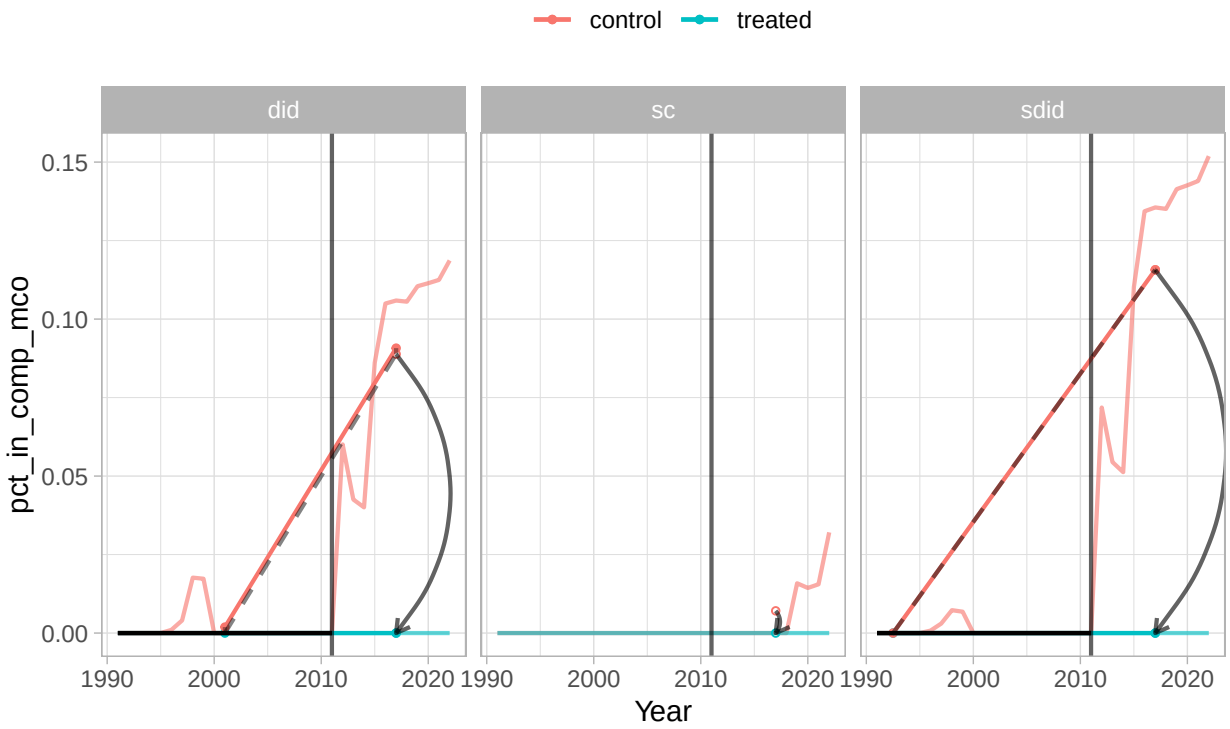

```
## Plot saved for cohort year 2007
```



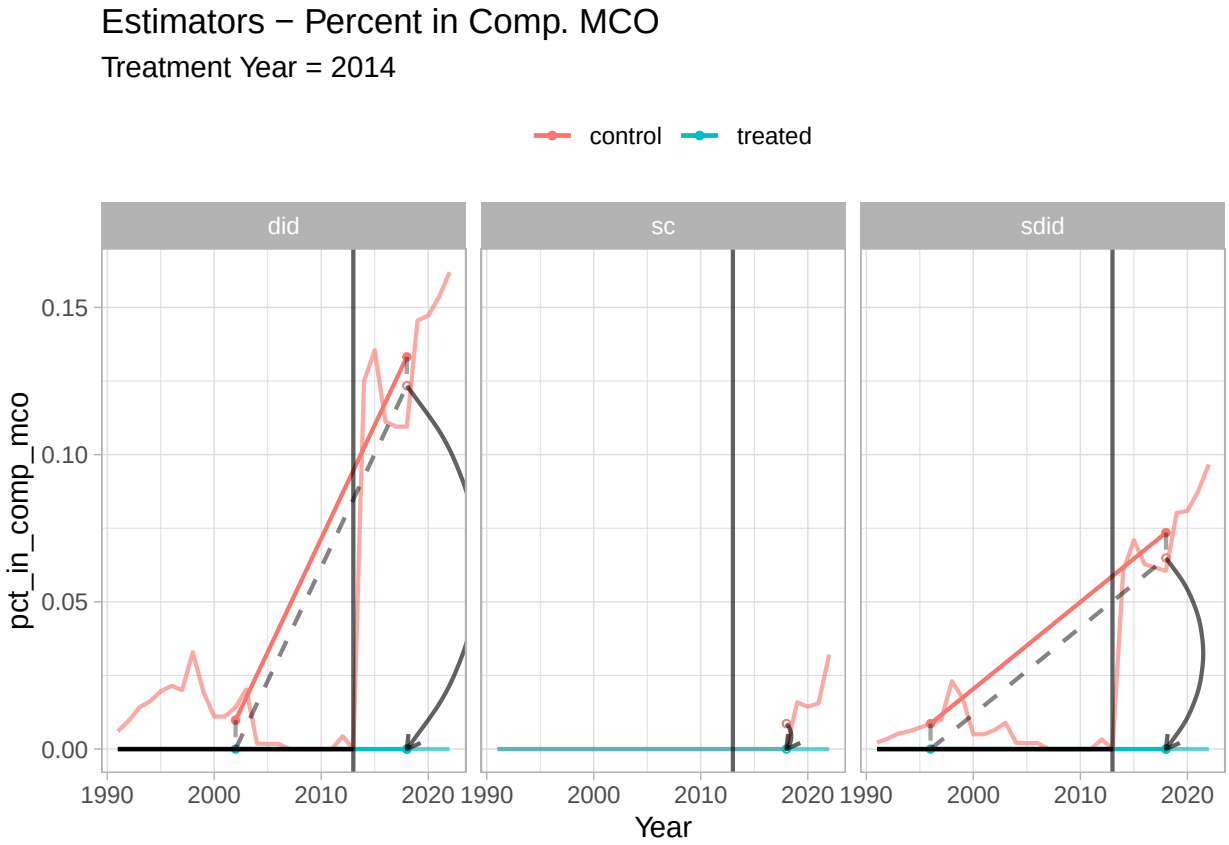

```
## Plot saved for cohort year 2011
```



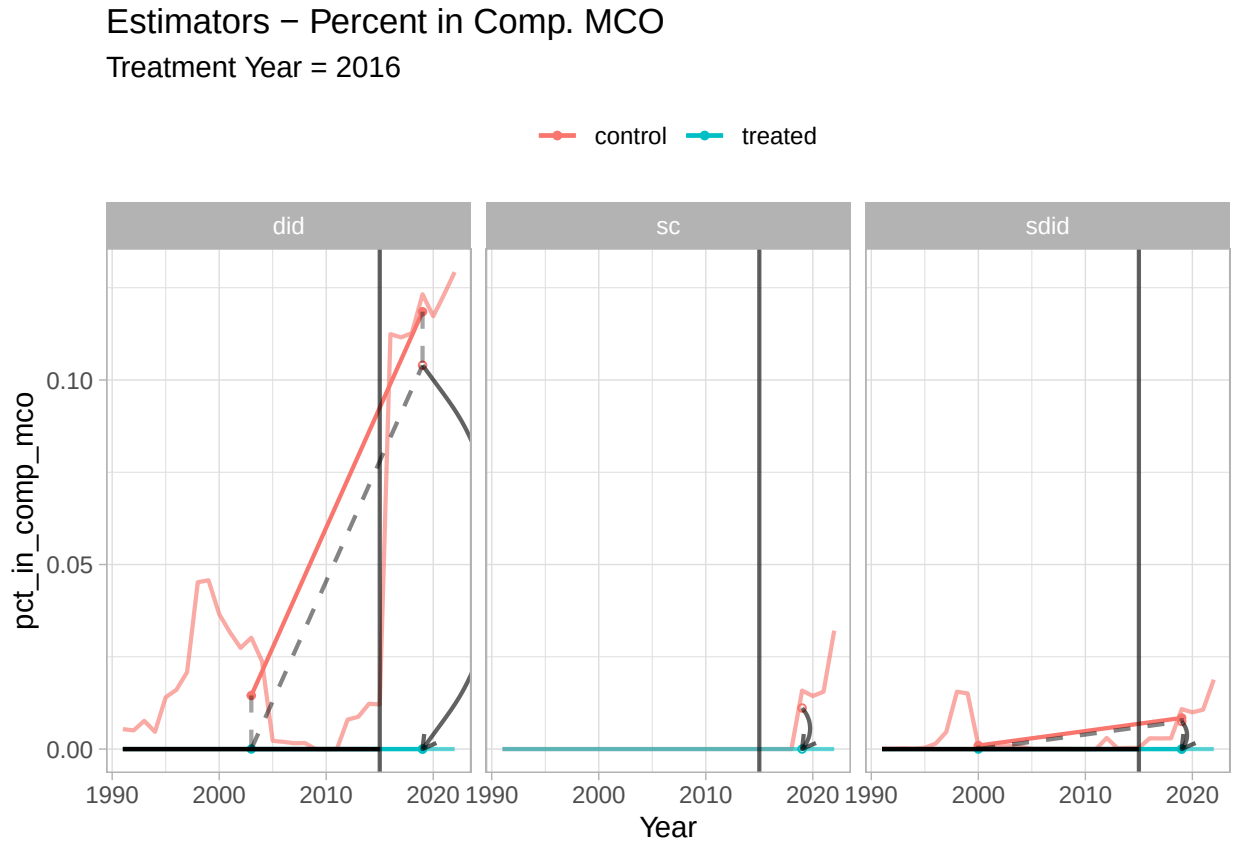

```
## Plot saved for cohort year 2012
```



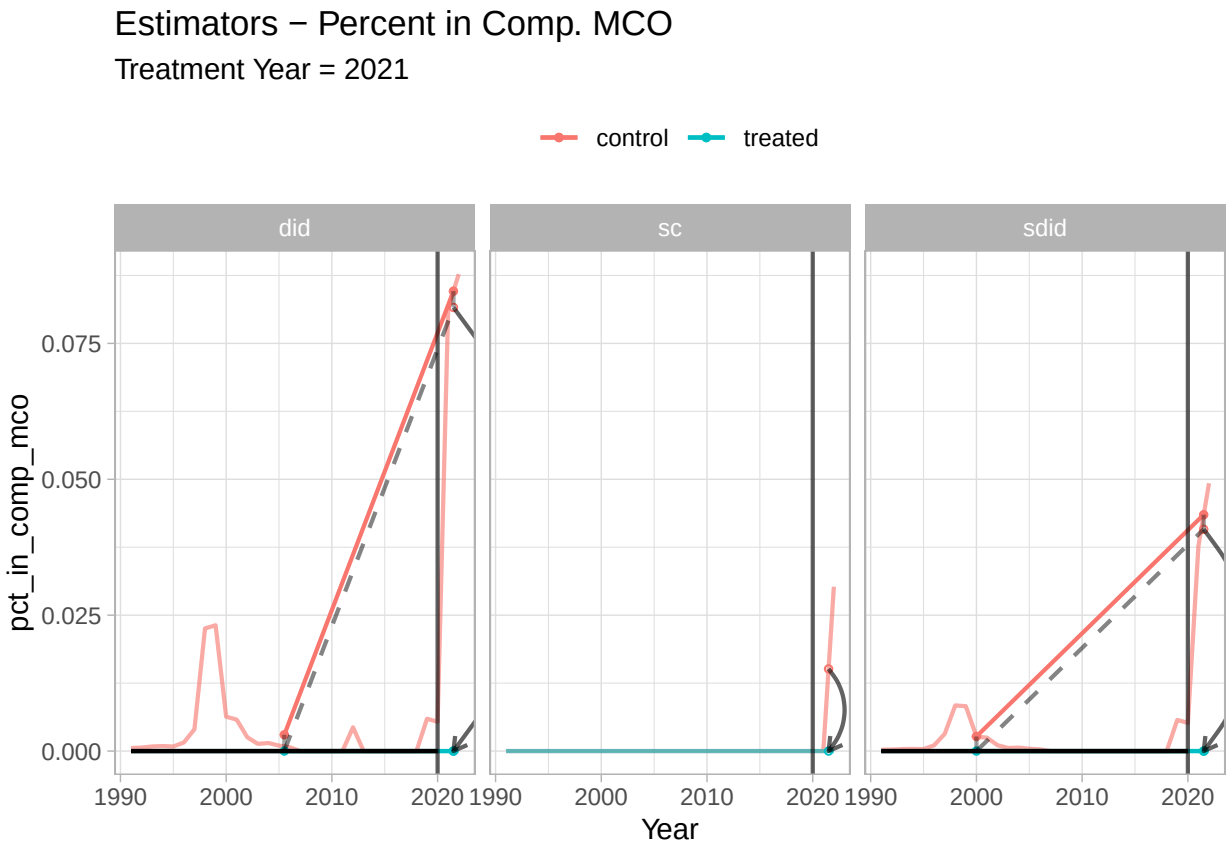

```
## Plot saved for cohort year 2014
```




```
## Plot saved for cohort year 2016
```




```
## Plot saved for cohort year 2021
```

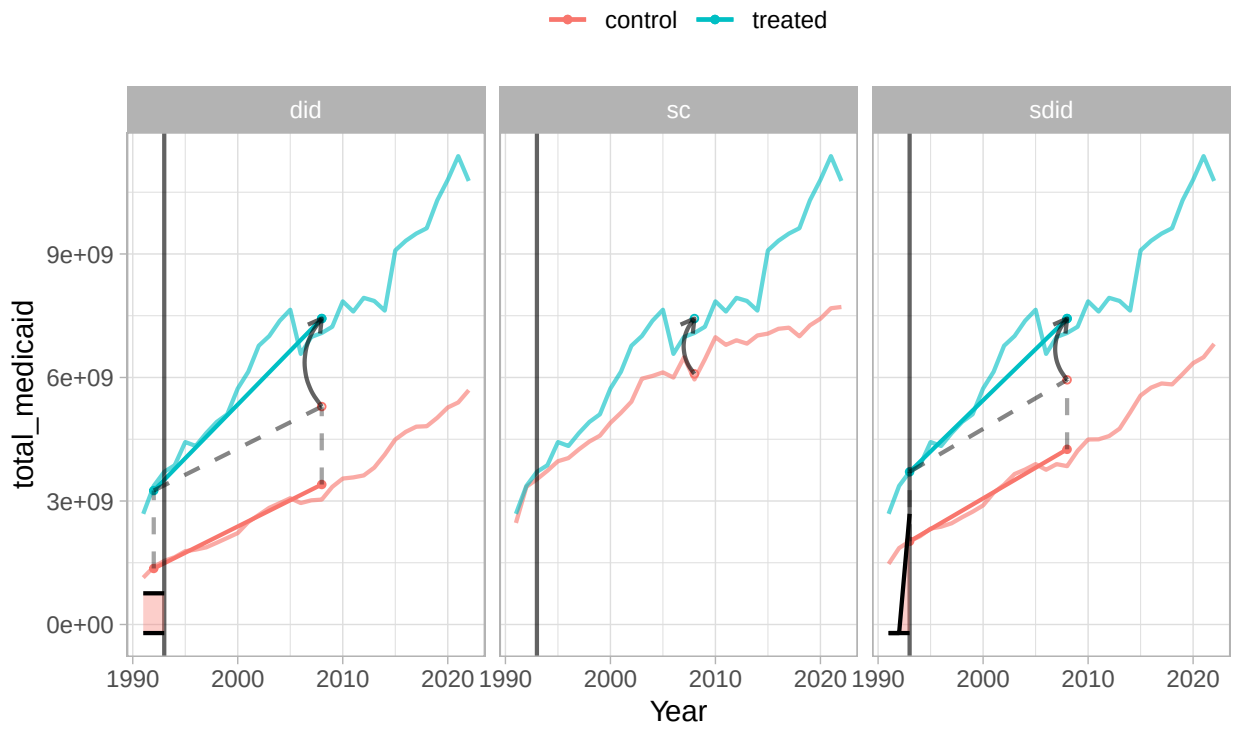


Total Medicaid Spending

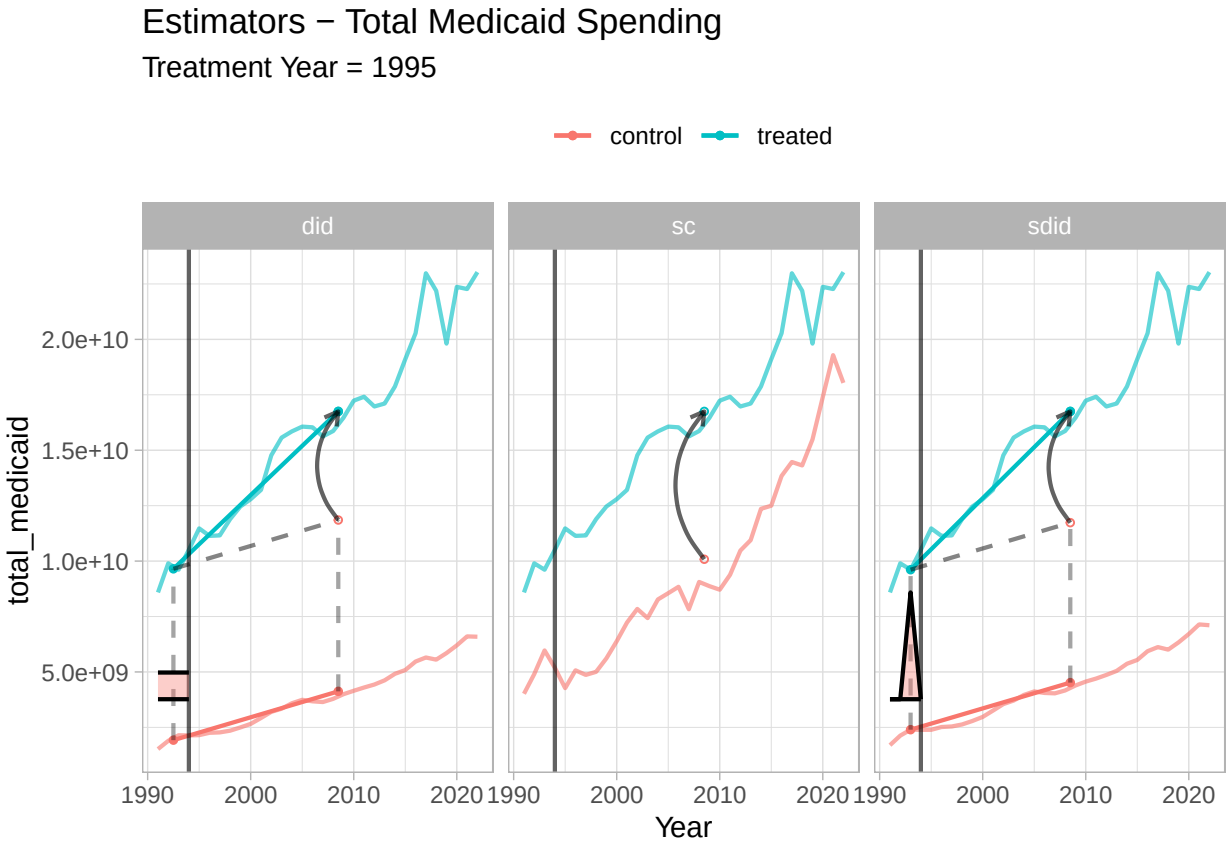
```
run_synthdid(data, "total_medicaid", cohorts)
```

```
## Error in cohort year 1992: dim(X) must have a positive length
```

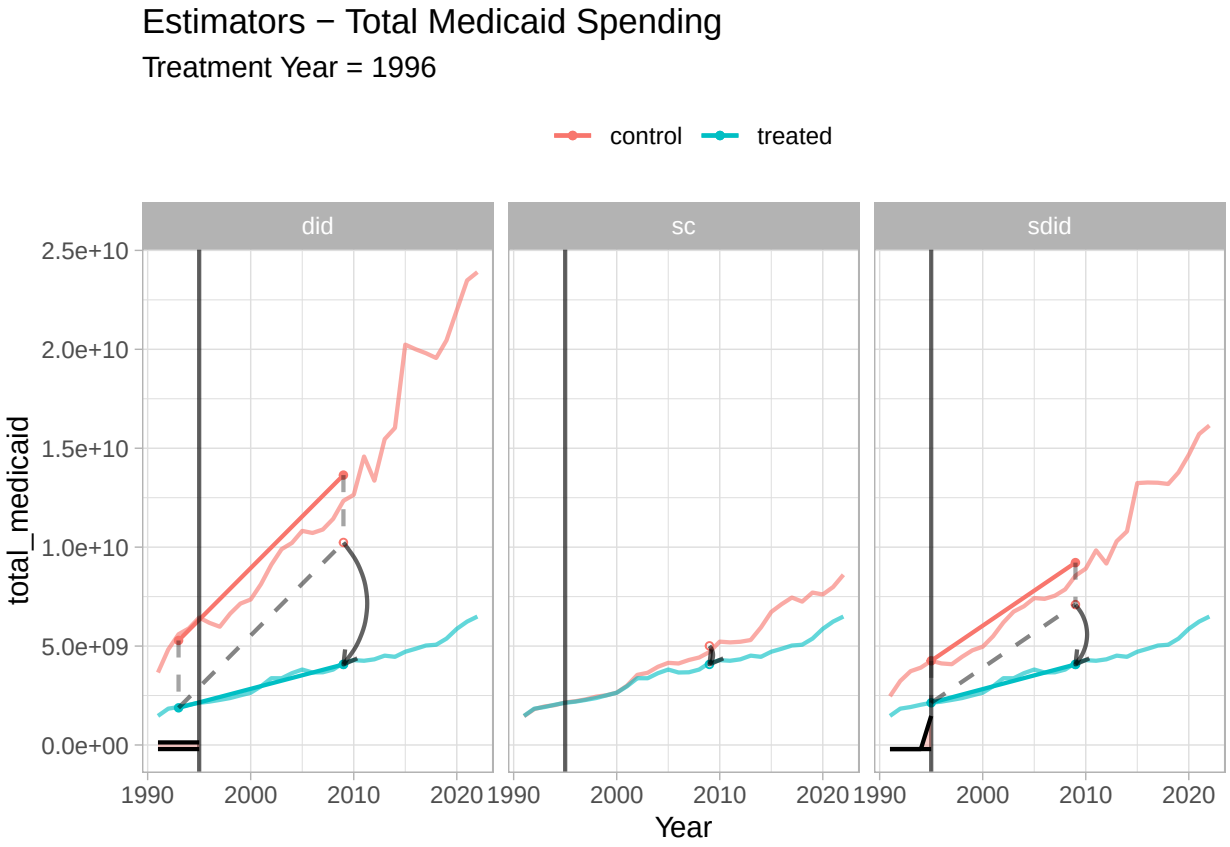
```
## Plot saved for cohort year 1994
```



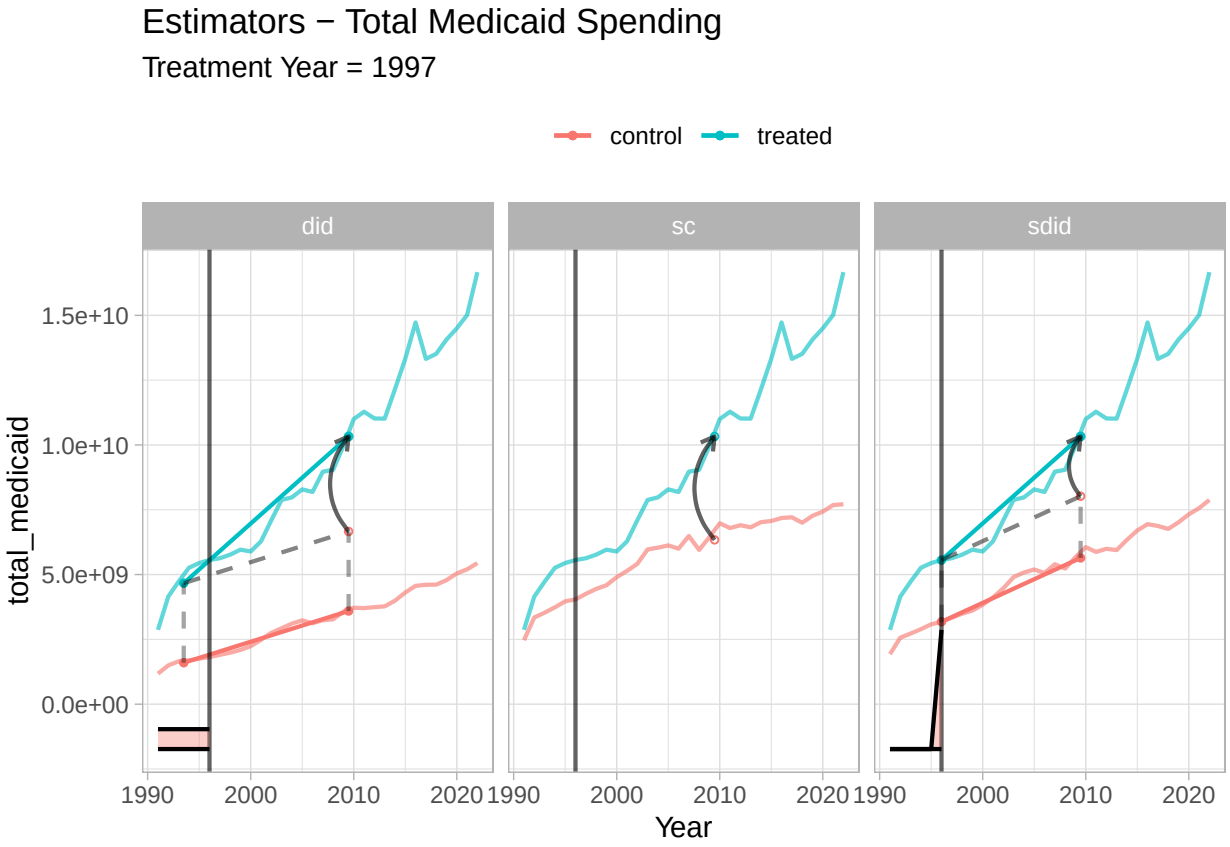

```
## Plot saved for cohort year 1995
```



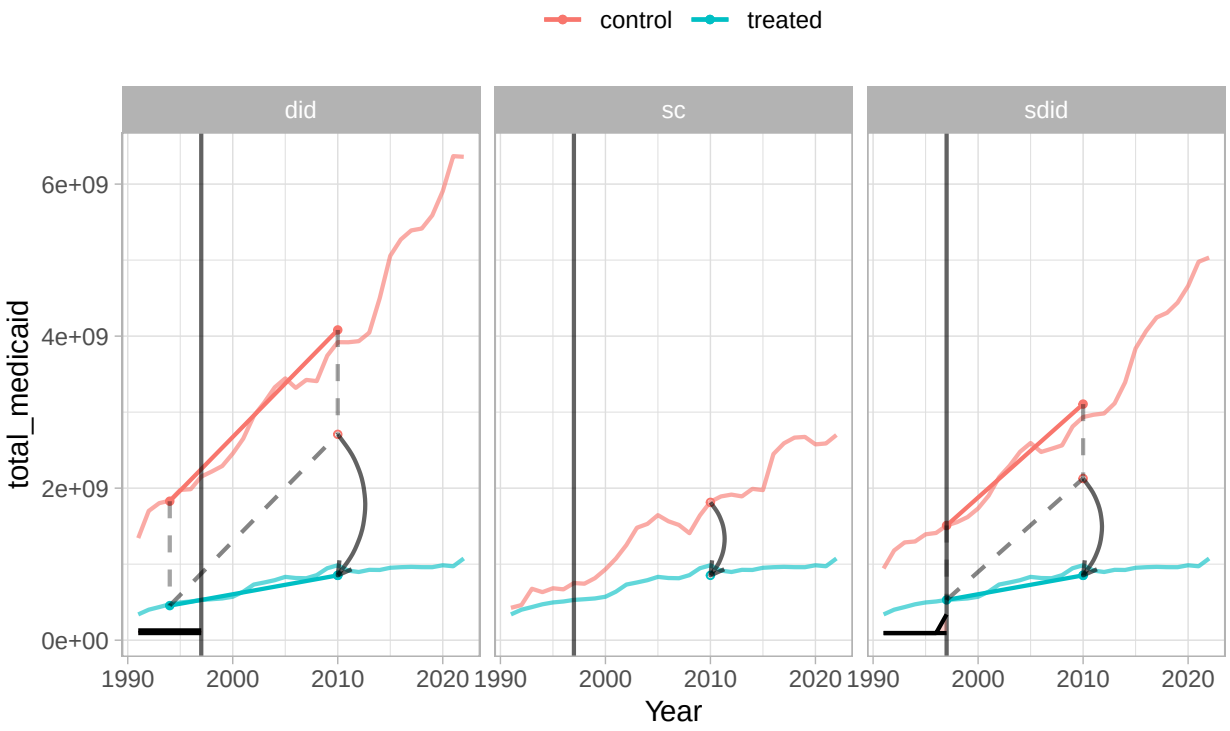

```
## Plot saved for cohort year 1996
```



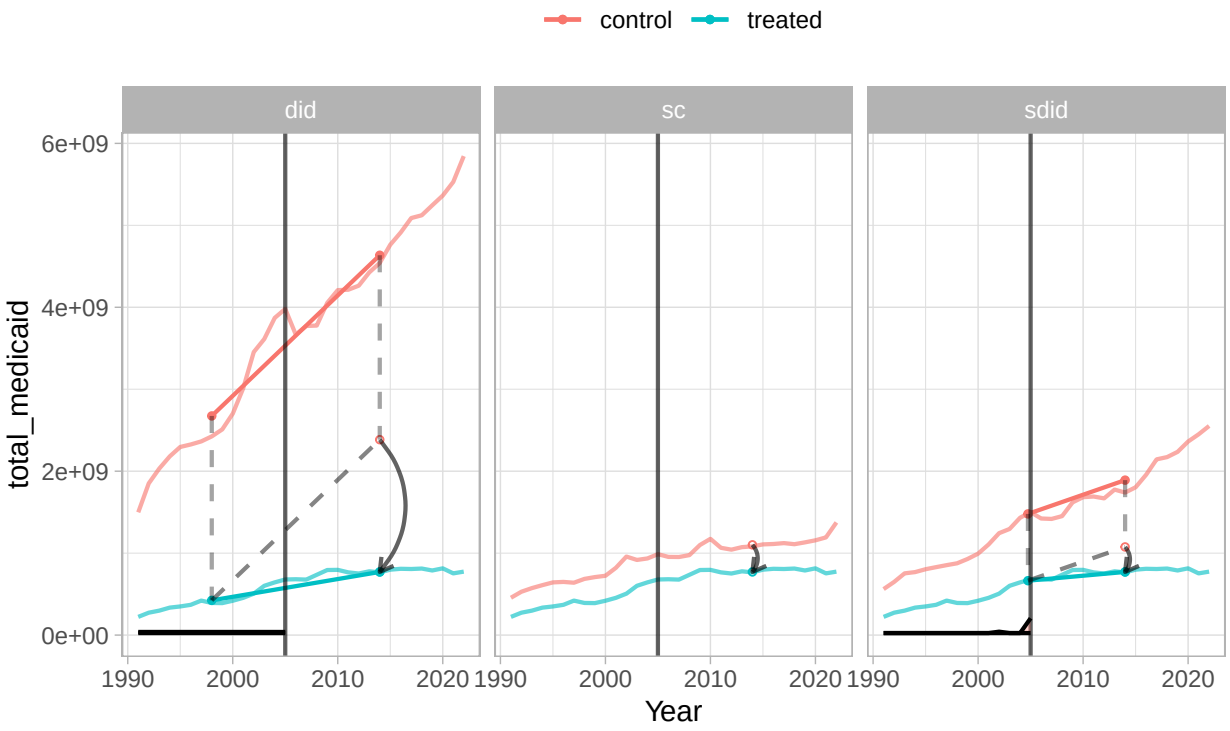

```
## Plot saved for cohort year 1997
```



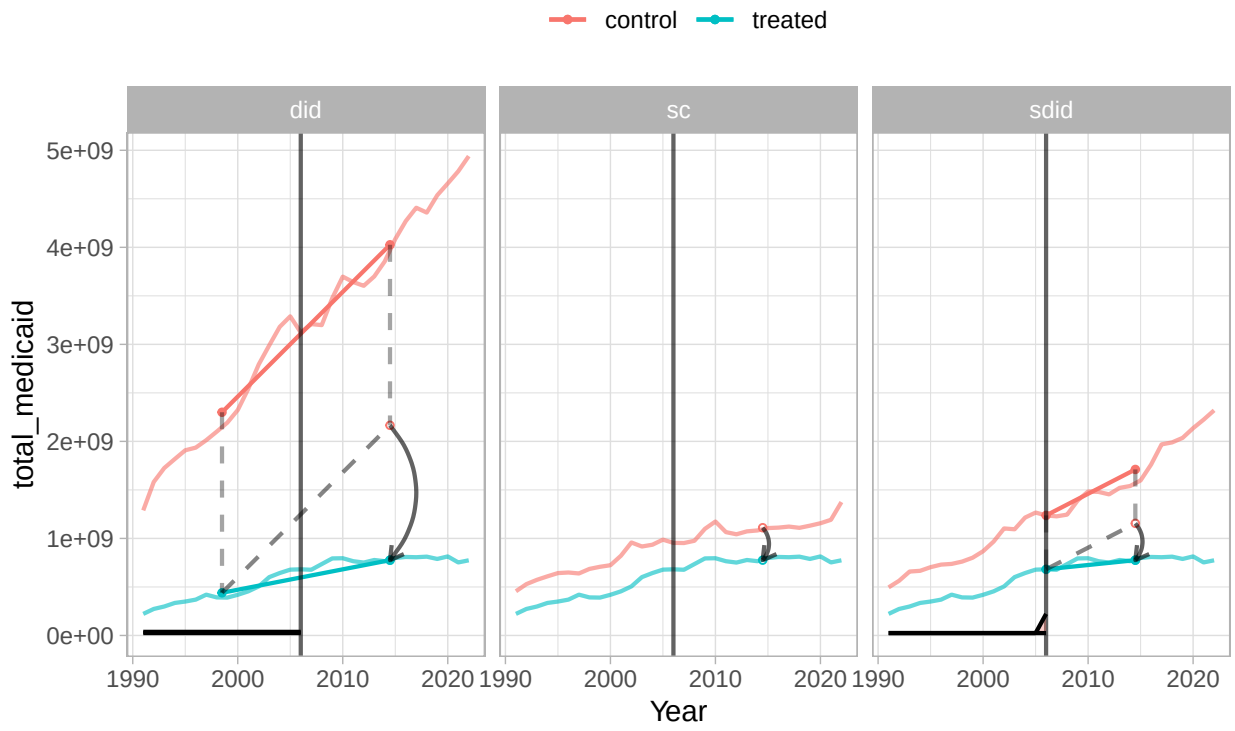

```
## Plot saved for cohort year 1998
```



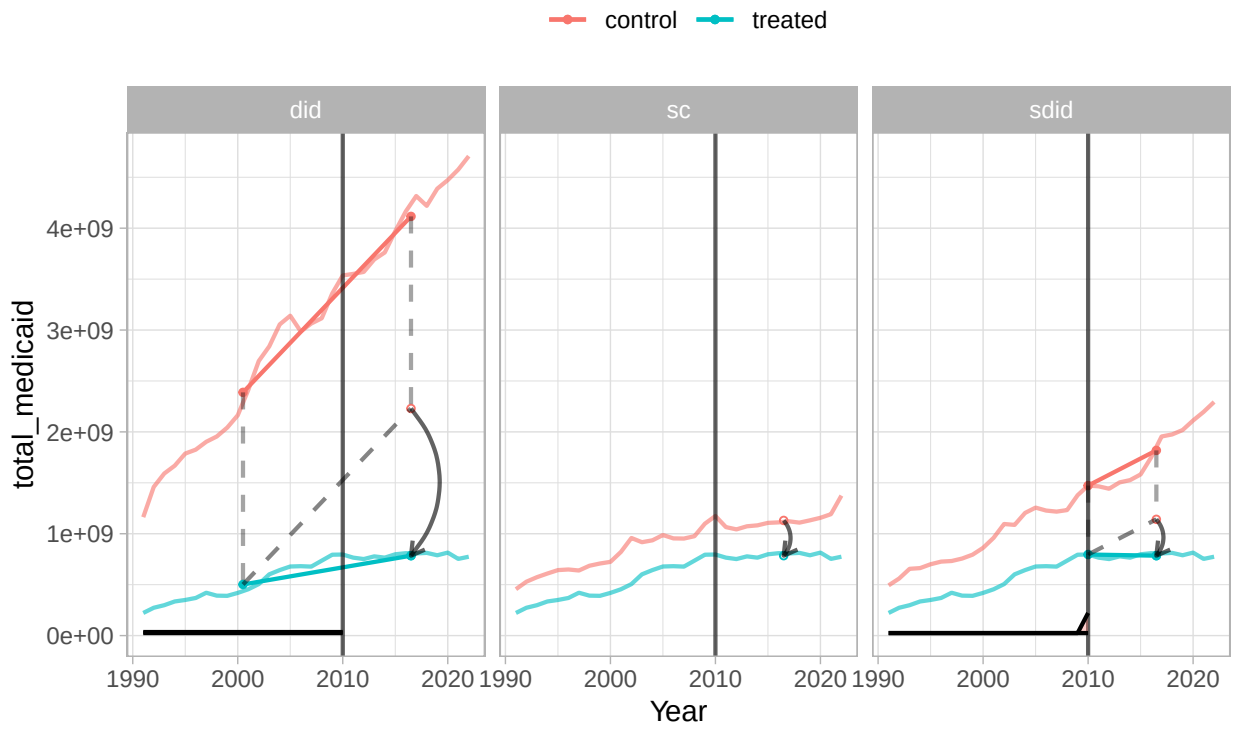

```
## Plot saved for cohort year 2006
```



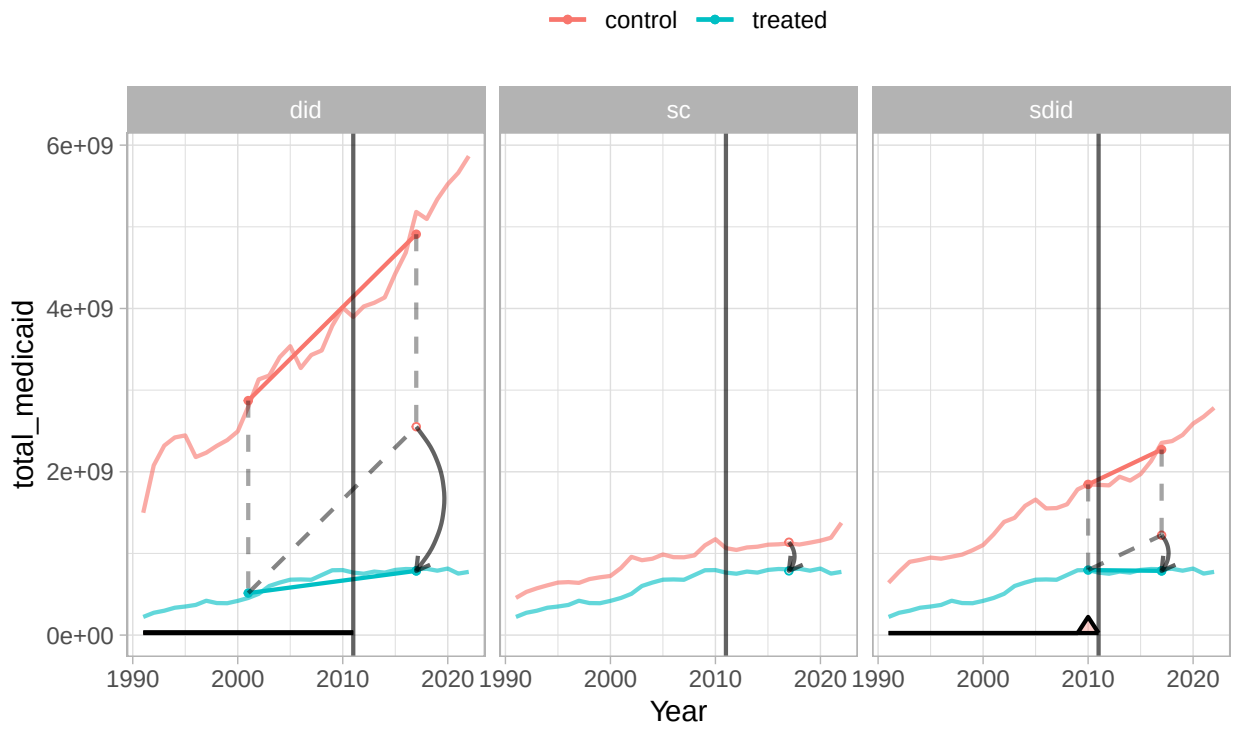

```
## Plot saved for cohort year 2007
```



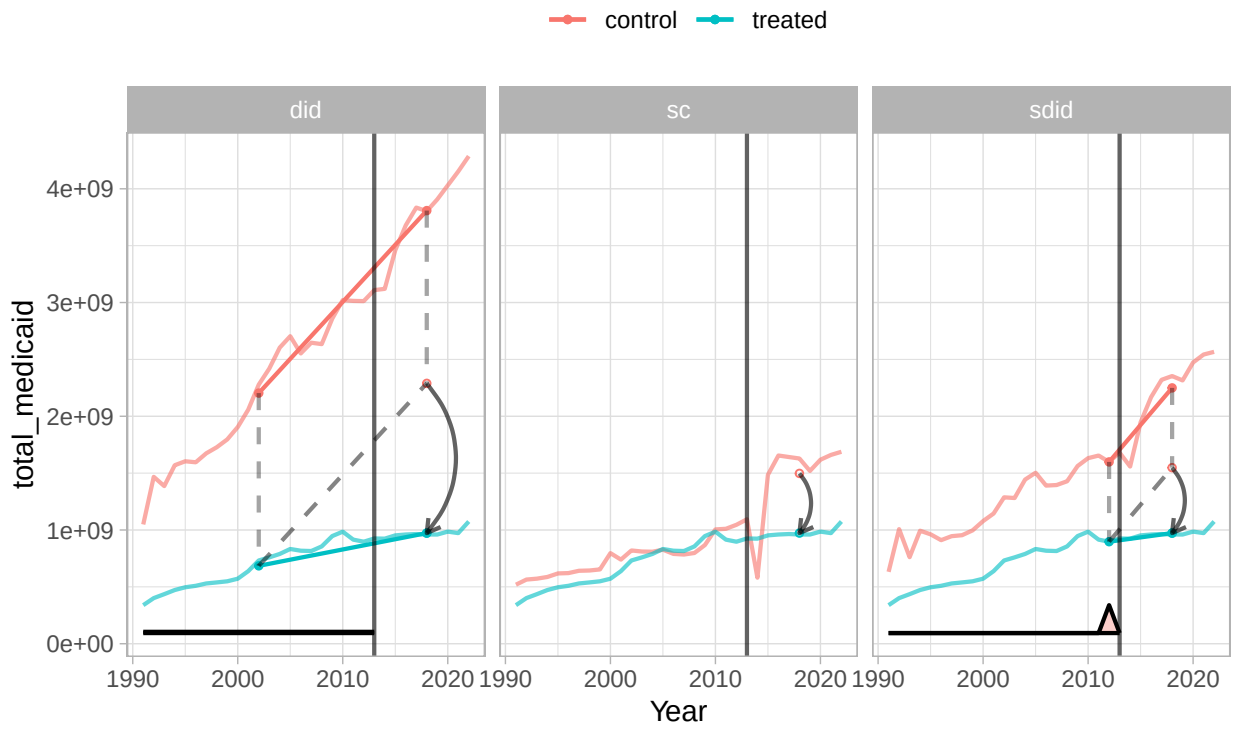

```
## Plot saved for cohort year 2011
```



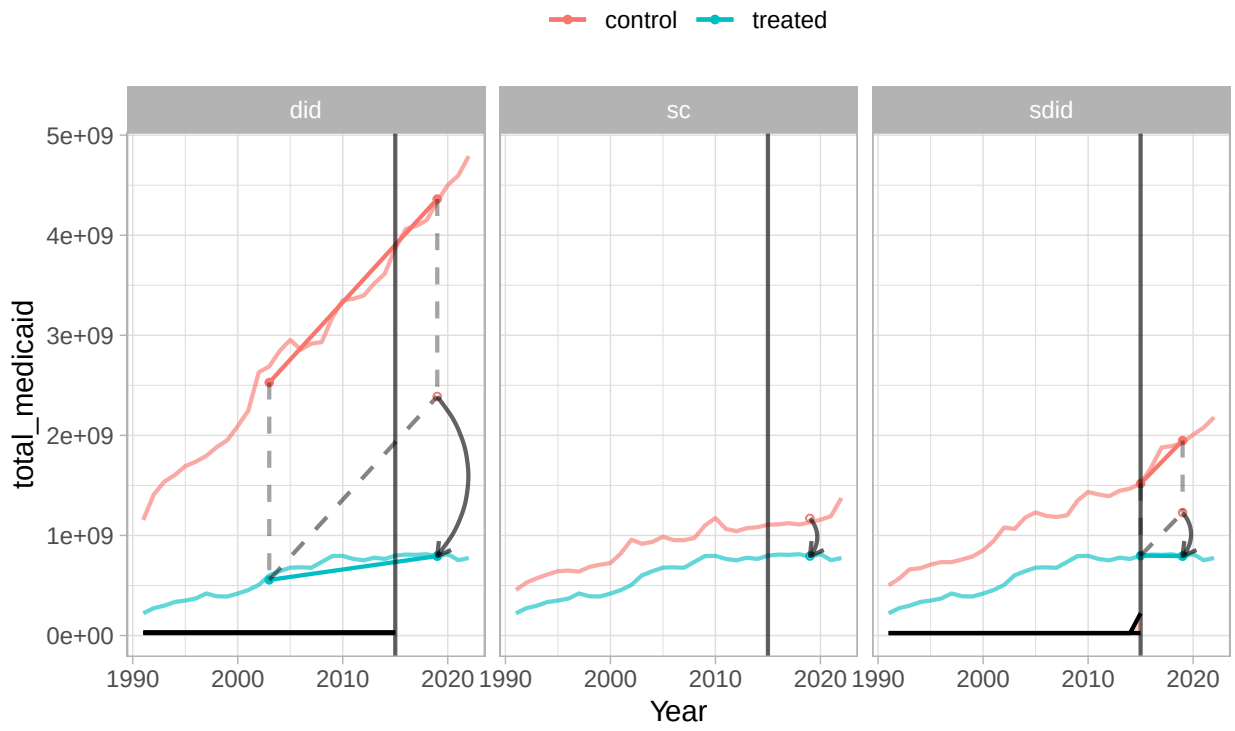

```
## Plot saved for cohort year 2012
```



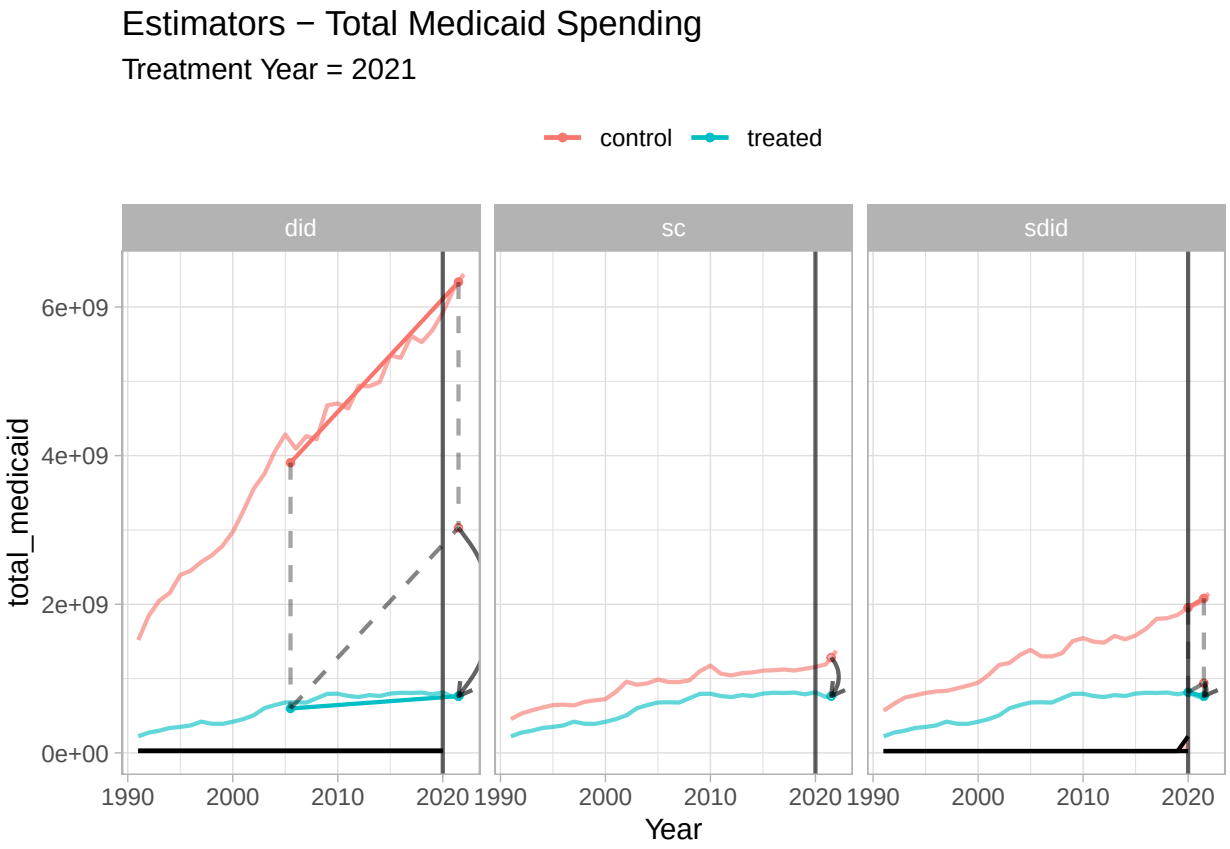

```
## Plot saved for cohort year 2014
```




```
## Plot saved for cohort year 2016
```




```
## Plot saved for cohort year 2021
```



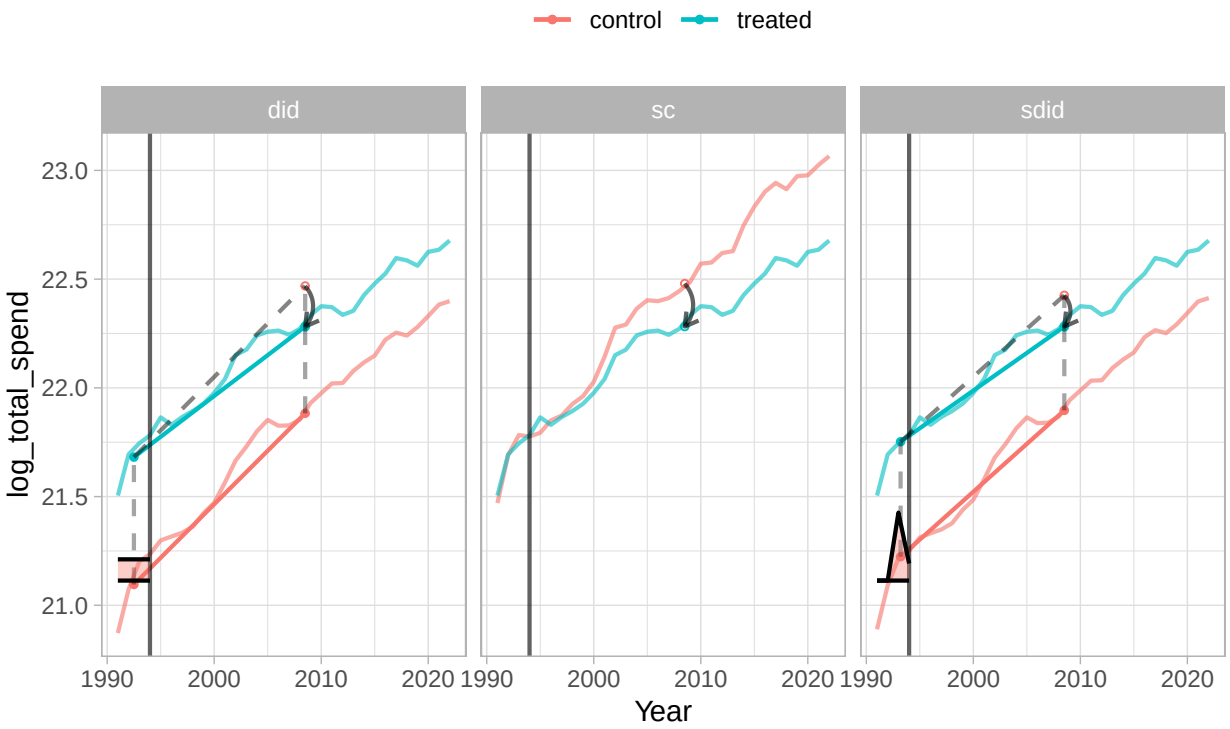
Log(Total Medicaid Spending)

```
run_synthdid(data, "log_total_spend", cohorts)
```

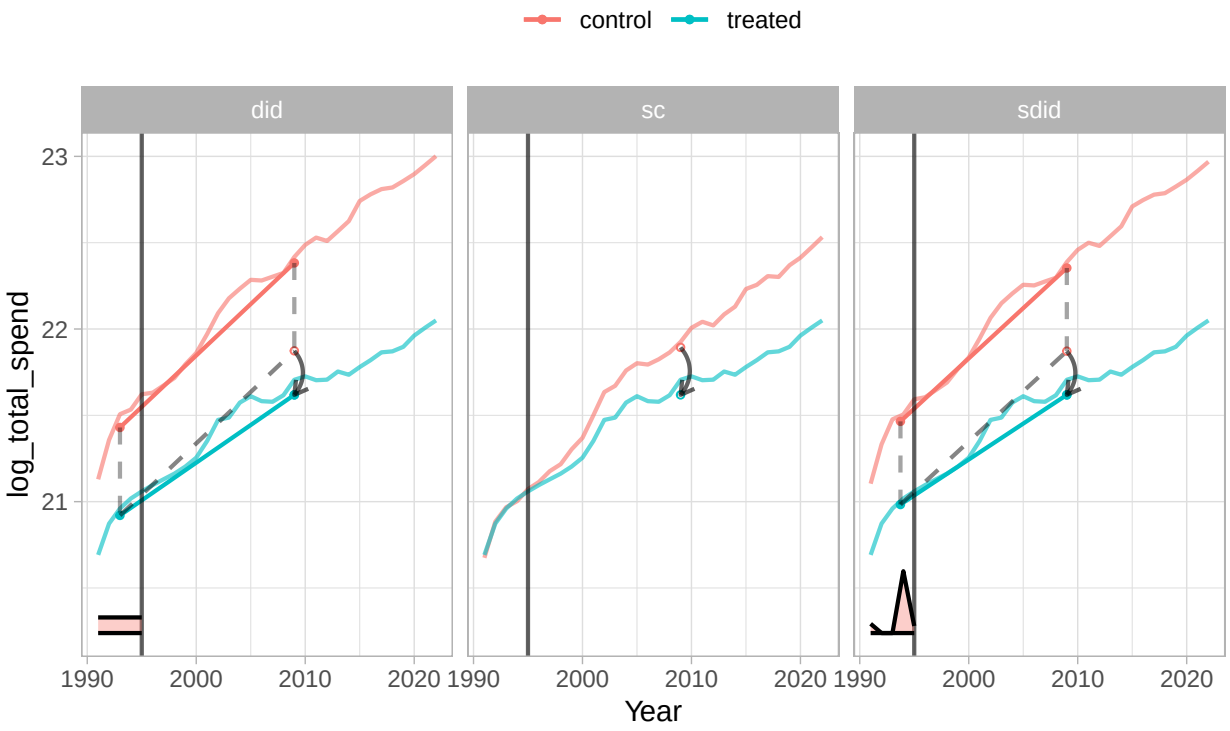
```
## Error in cohort year 1992: dim(X) must have a positive length
```

```
## Plot saved for cohort year 1994
```

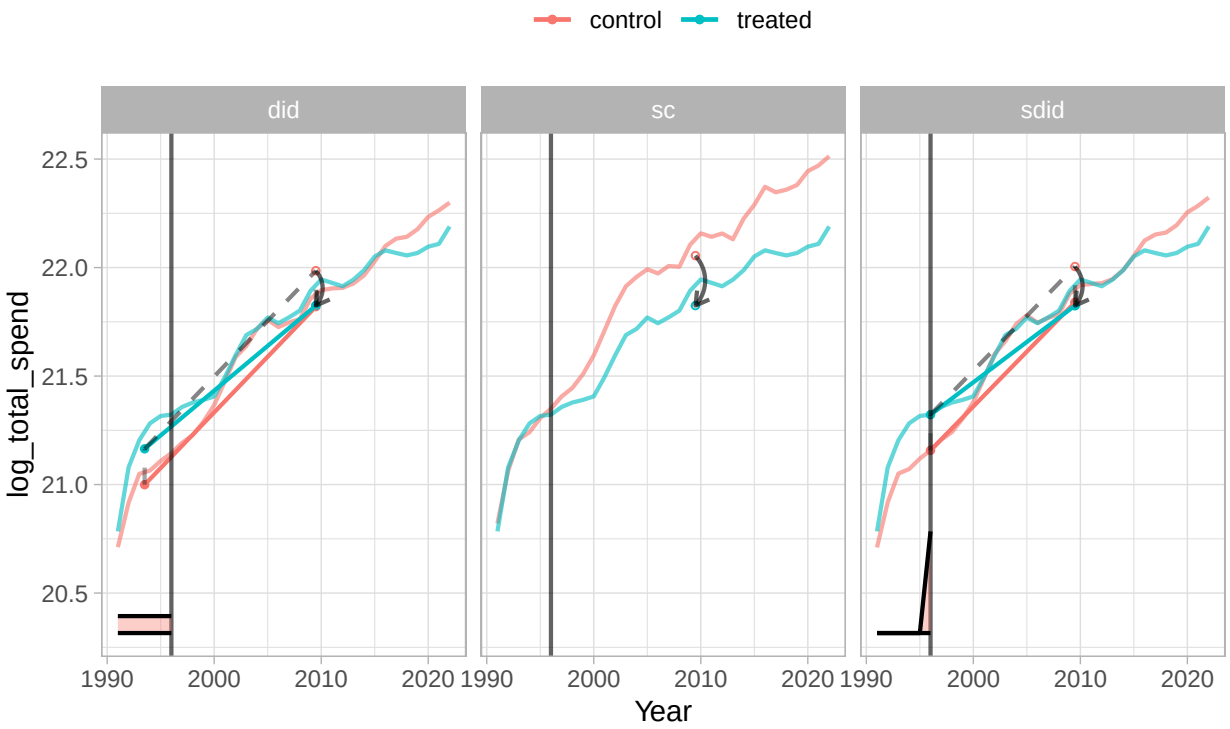

```
## Plot saved for cohort year 1995
```



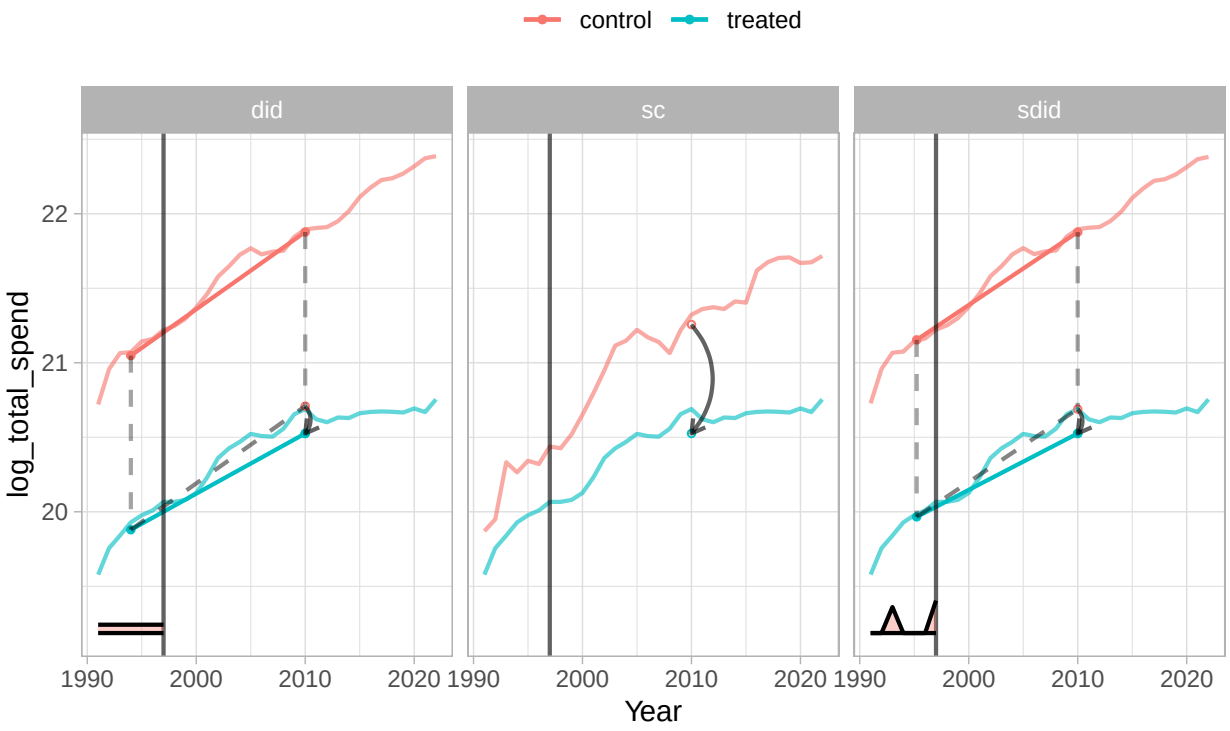

```
## Plot saved for cohort year 1996
```



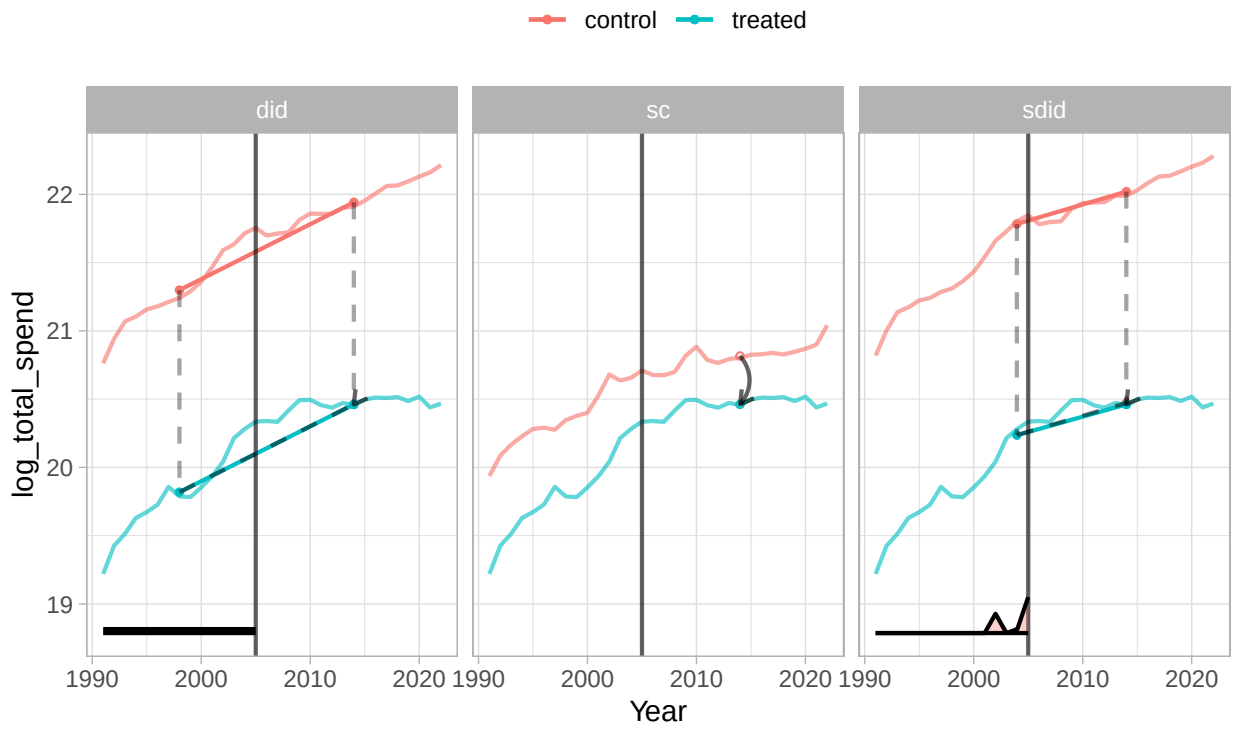

```
## Plot saved for cohort year 1997
```



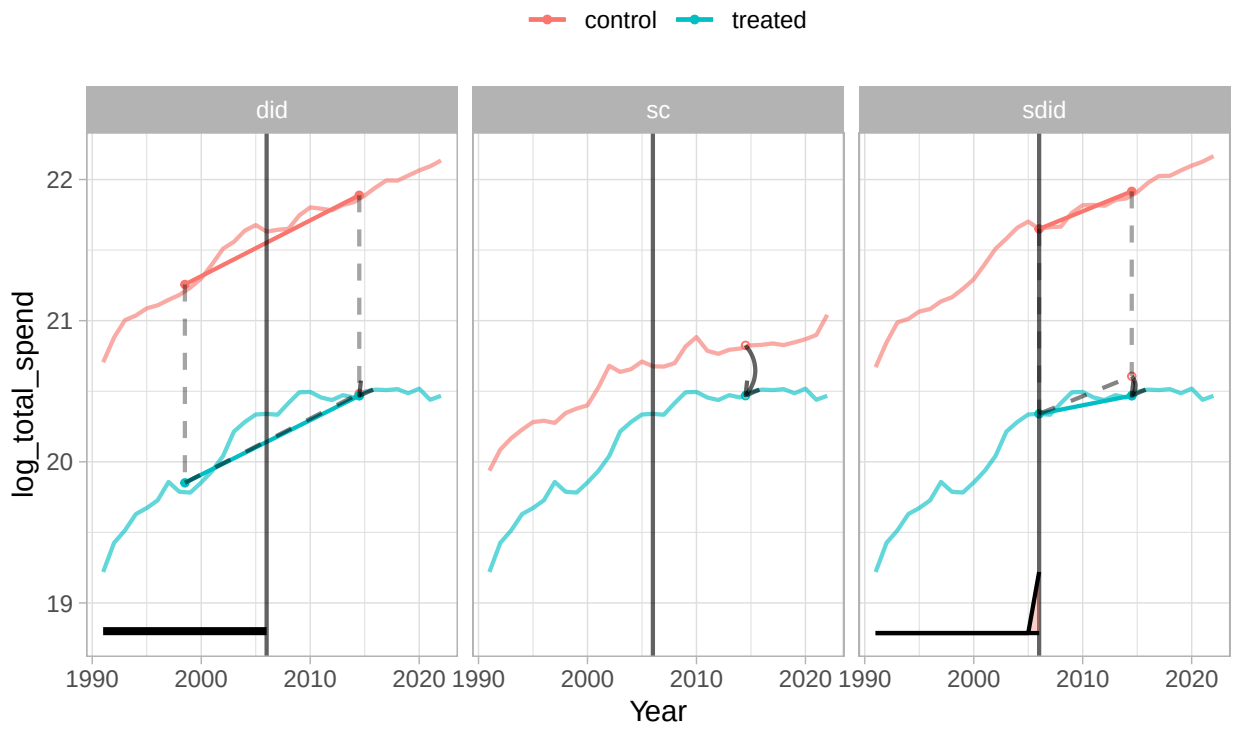

```
## Plot saved for cohort year 1998
```



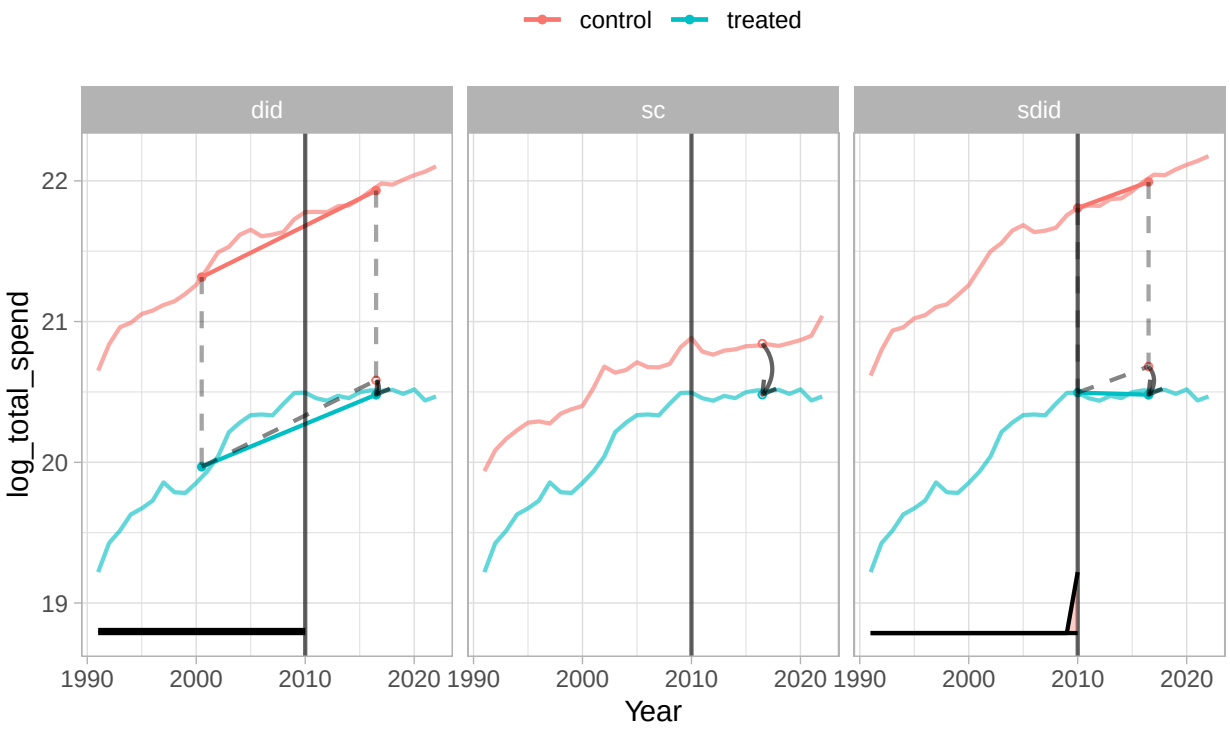

```
## Plot saved for cohort year 2006
```



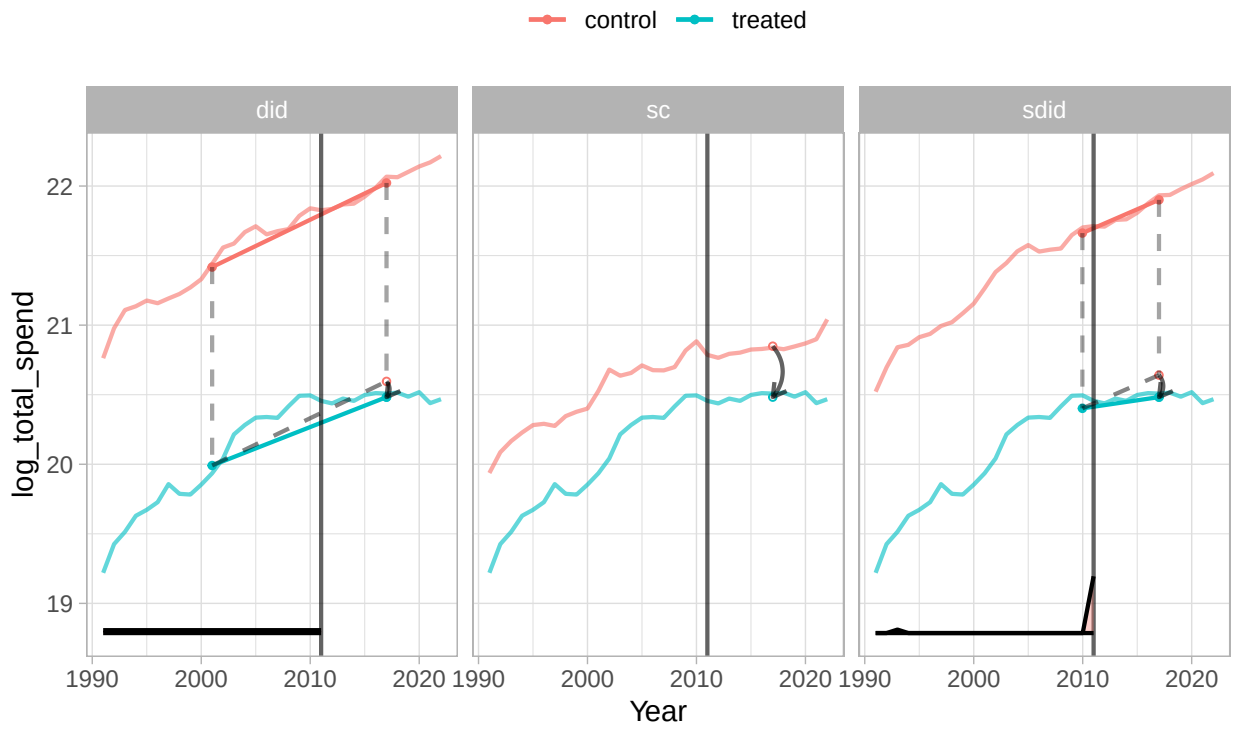

```
## Plot saved for cohort year 2007
```



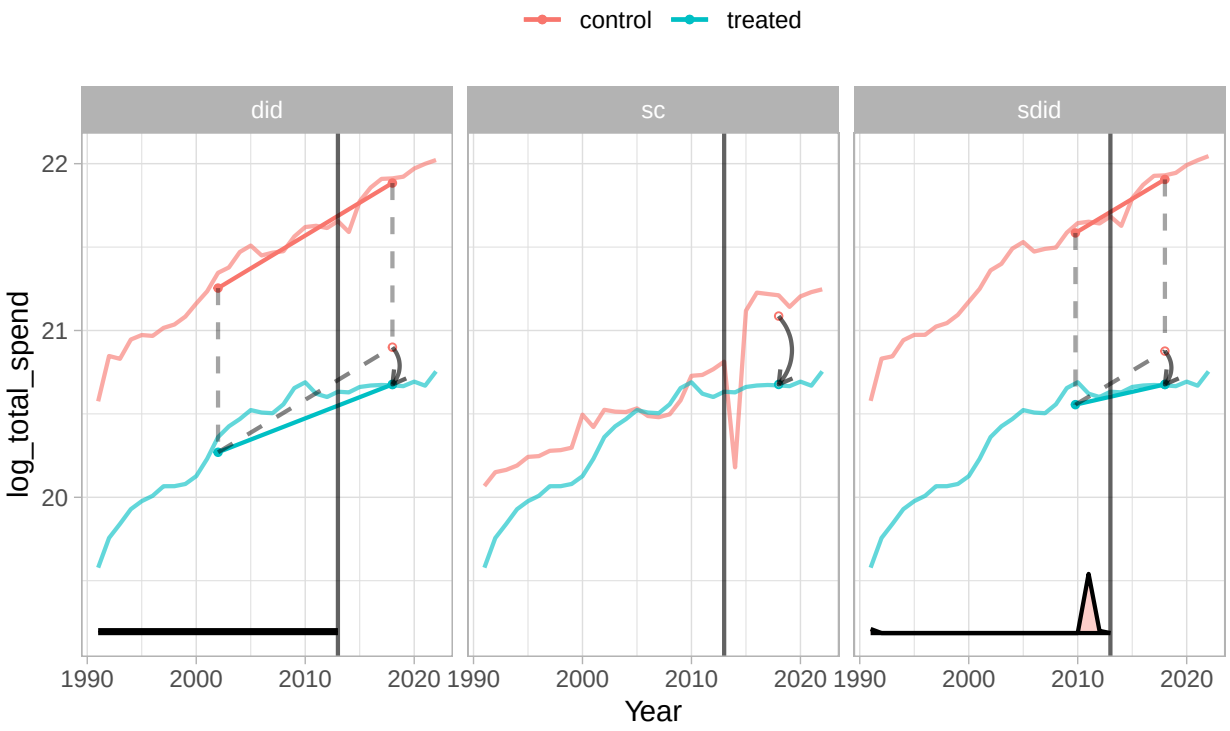

```
## Plot saved for cohort year 2011
```



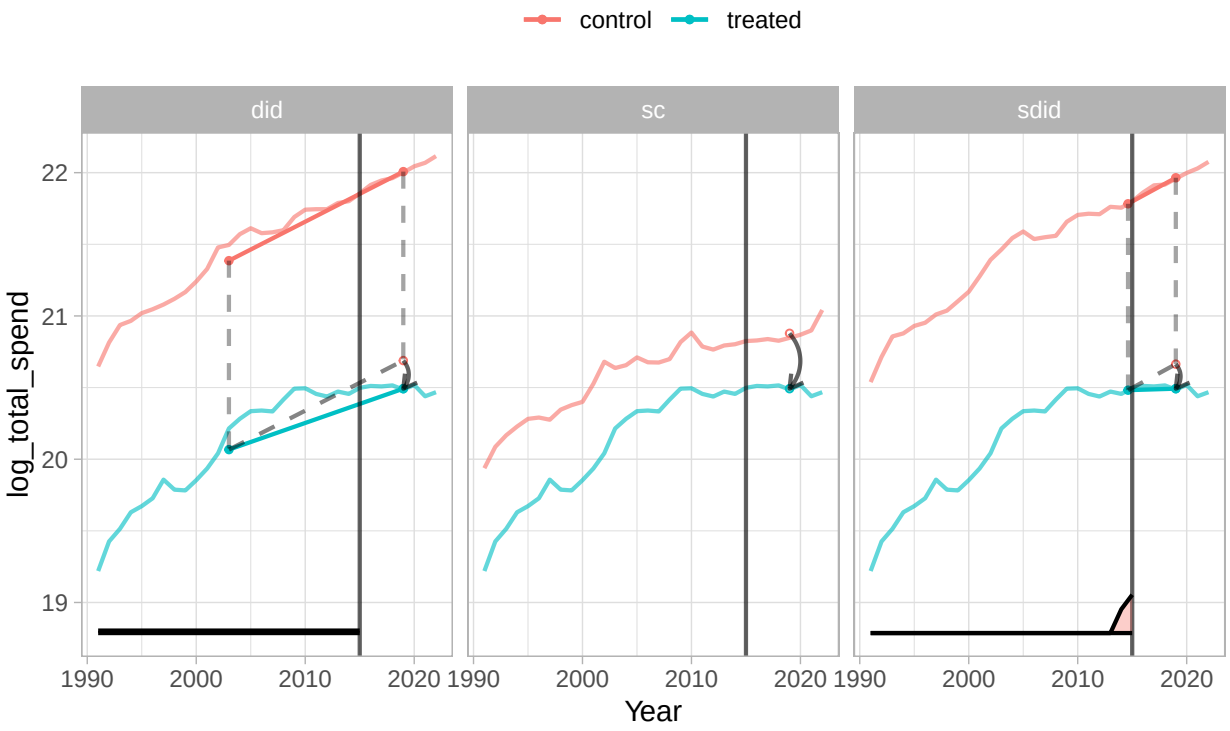

```
## Plot saved for cohort year 2012
```



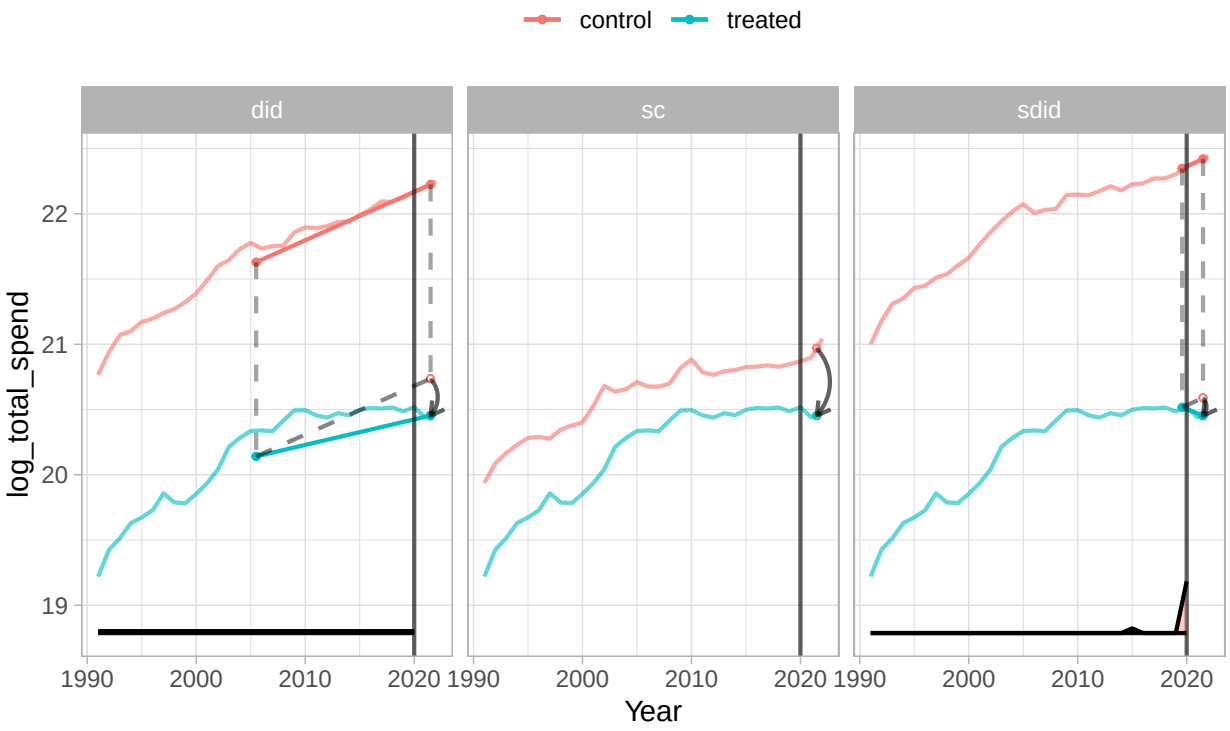

```
## Plot saved for cohort year 2014
```




```
## Plot saved for cohort year 2016
```




```
## Plot saved for cohort year 2021
```



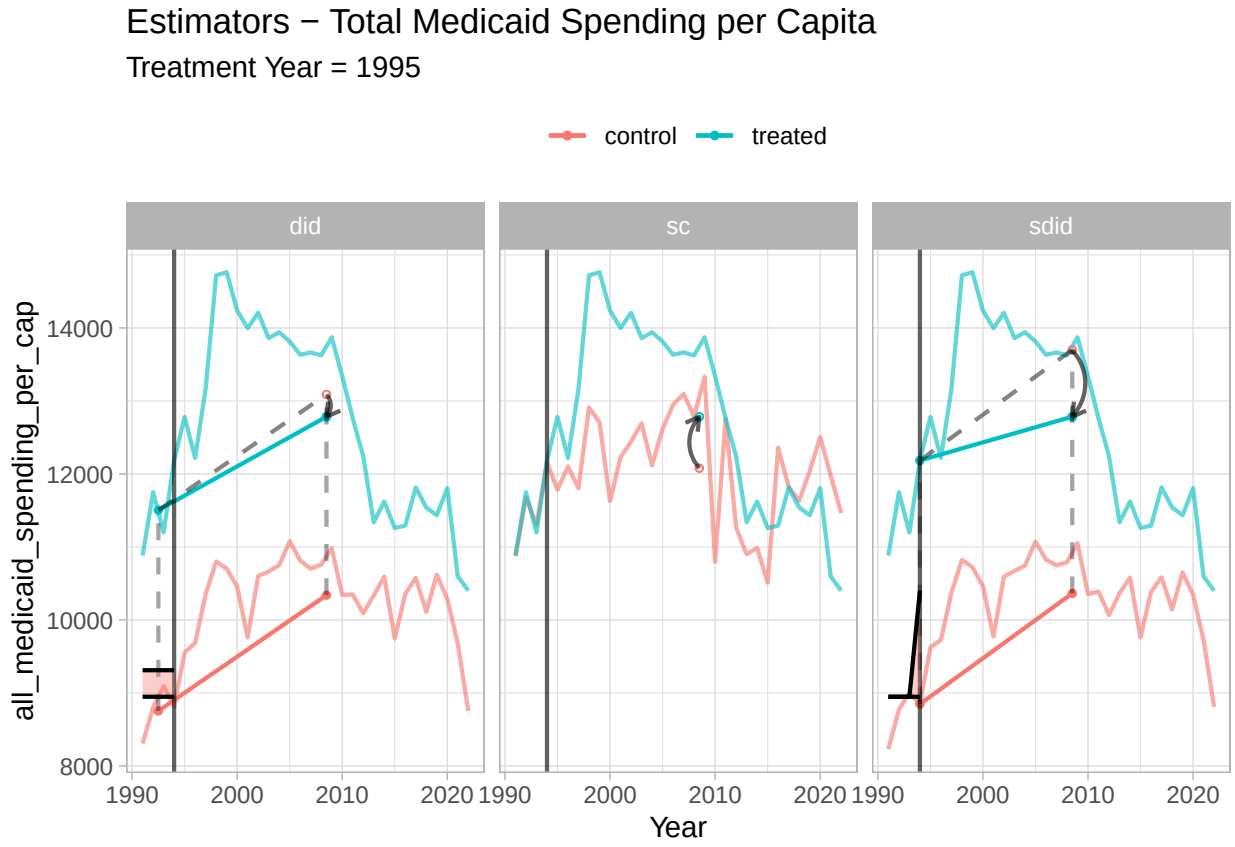
Total Medicaid Spending per Capita

```
run_synthdid(data, "all_medicaid_spending_per_cap", cohorts)
```

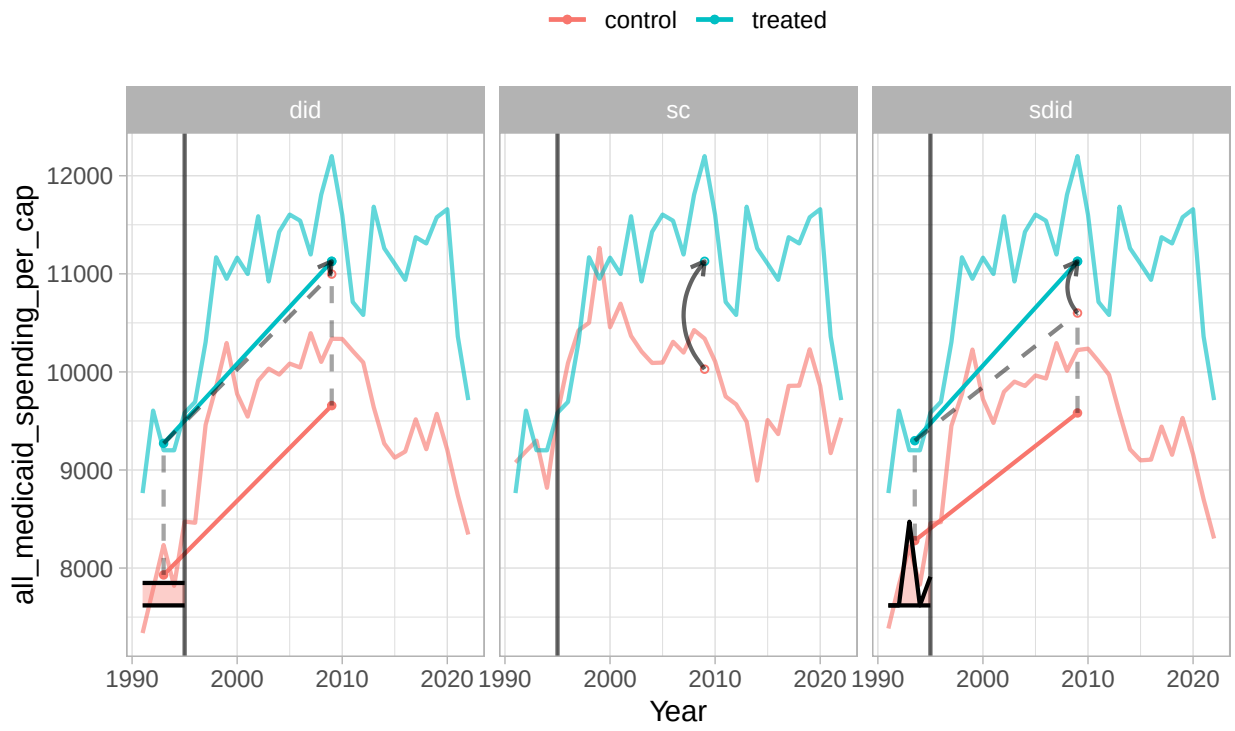
```
## Error in cohort year 1992: dim(X) must have a positive length
```

```
## Plot saved for cohort year 1994
```

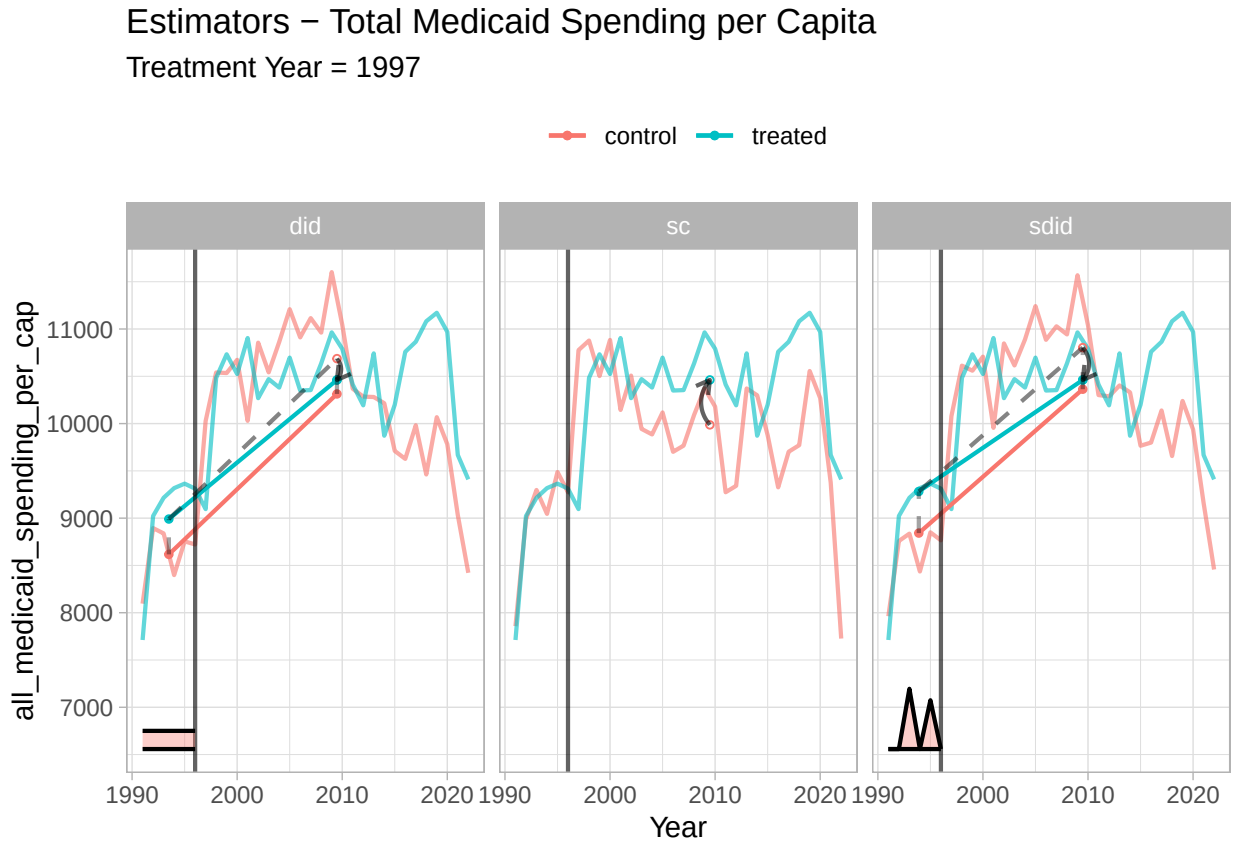

```
## Plot saved for cohort year 1995
```



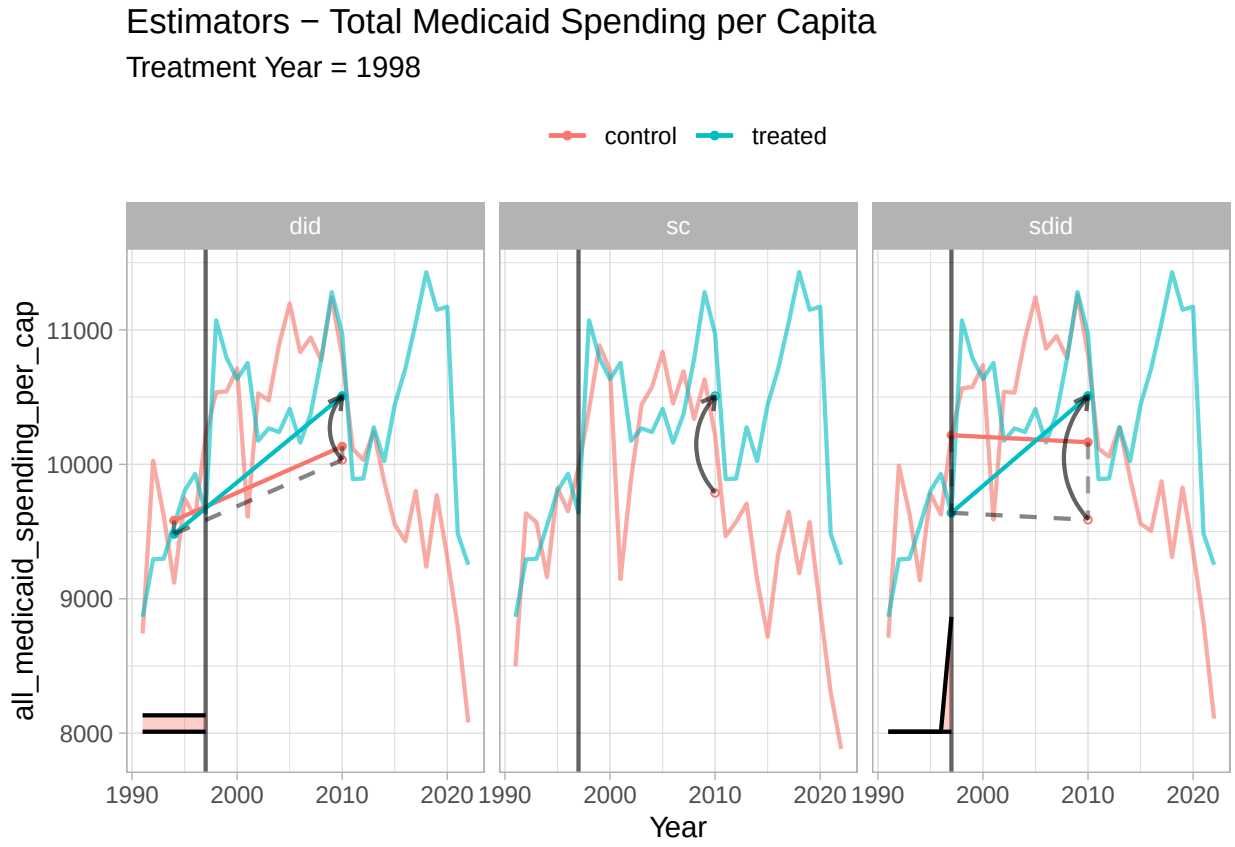

```
## Plot saved for cohort year 1996
```



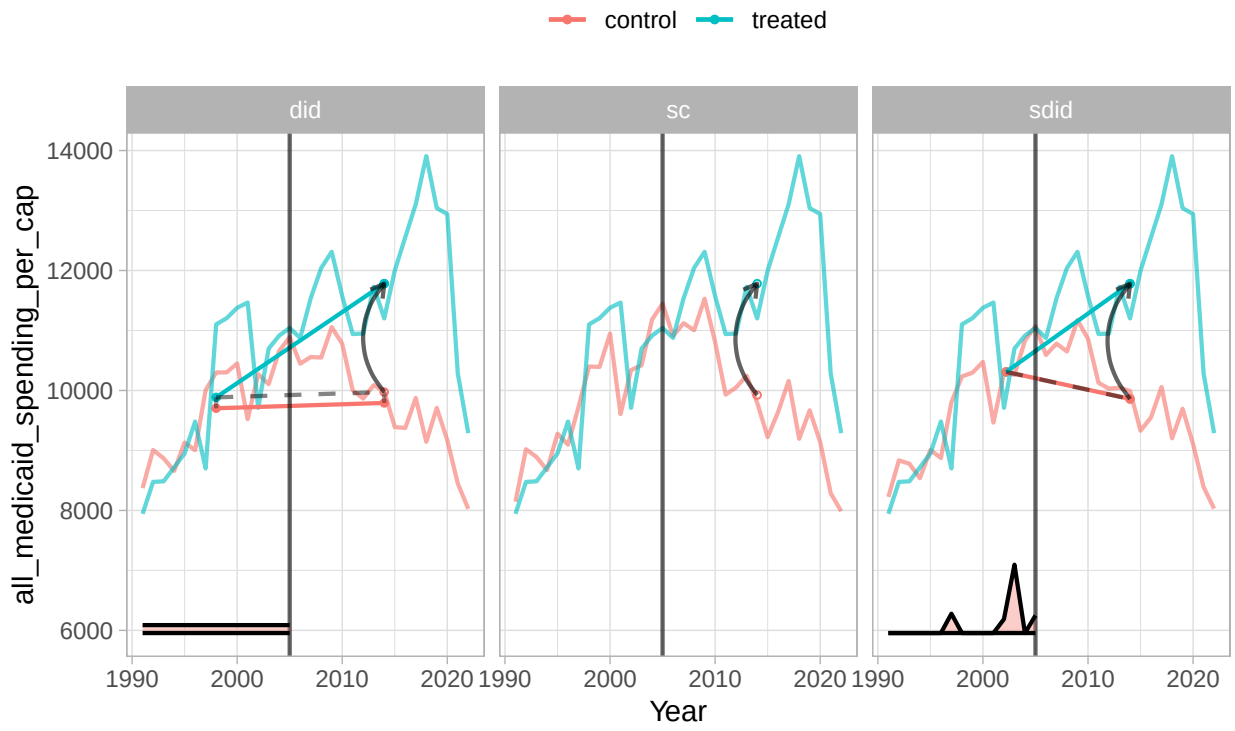

```
## Plot saved for cohort year 1997
```



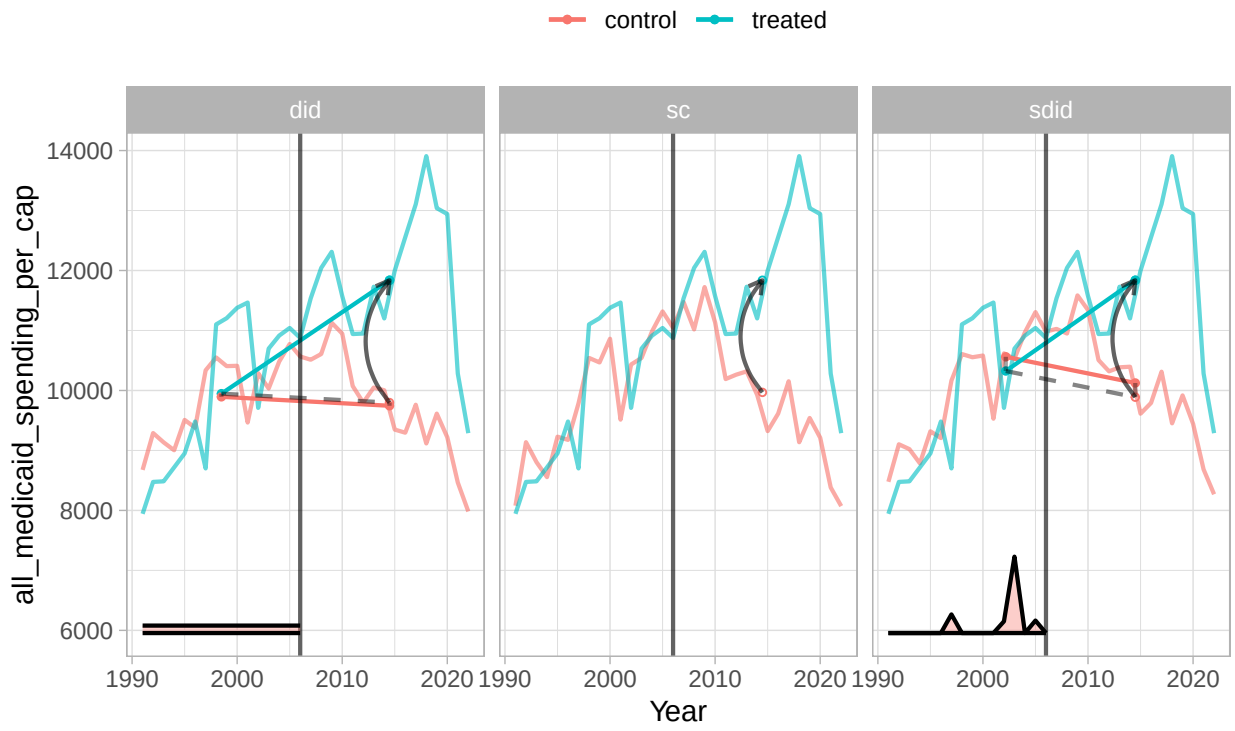

```
## Plot saved for cohort year 1998
```



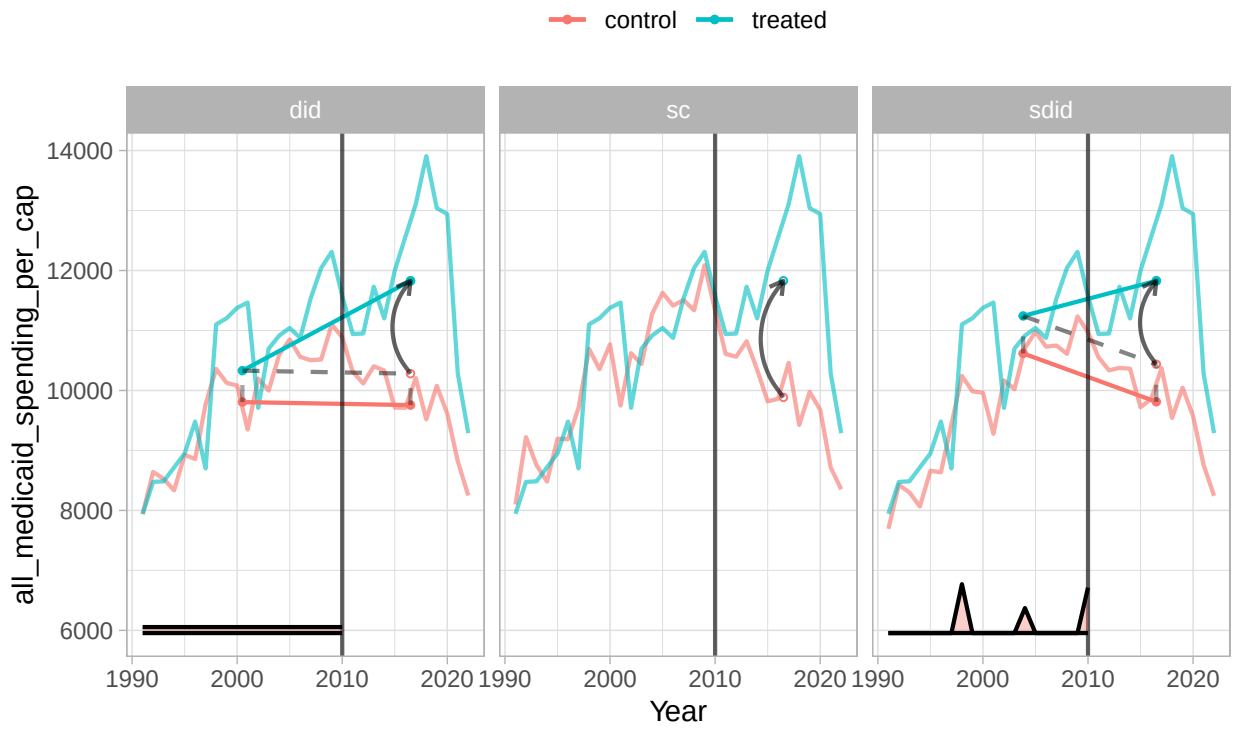

```
## Plot saved for cohort year 2006
```



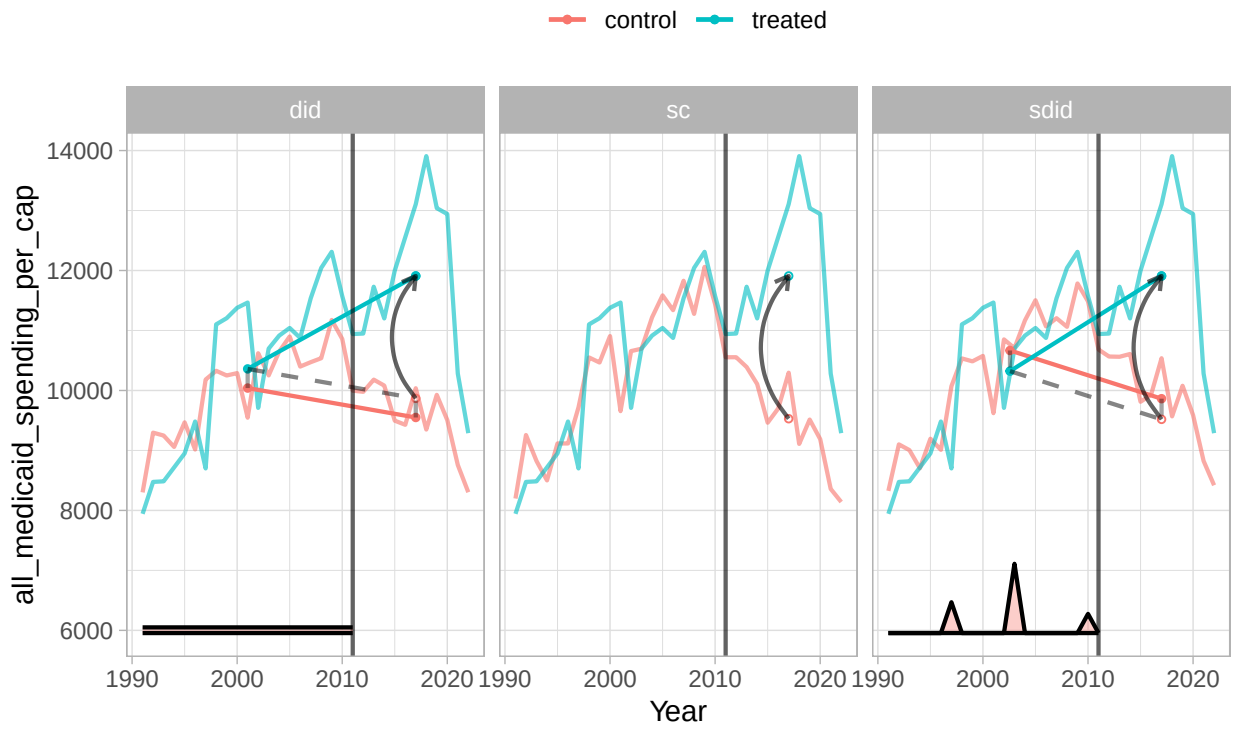

```
## Plot saved for cohort year 2007
```




```
## Plot saved for cohort year 2011
```

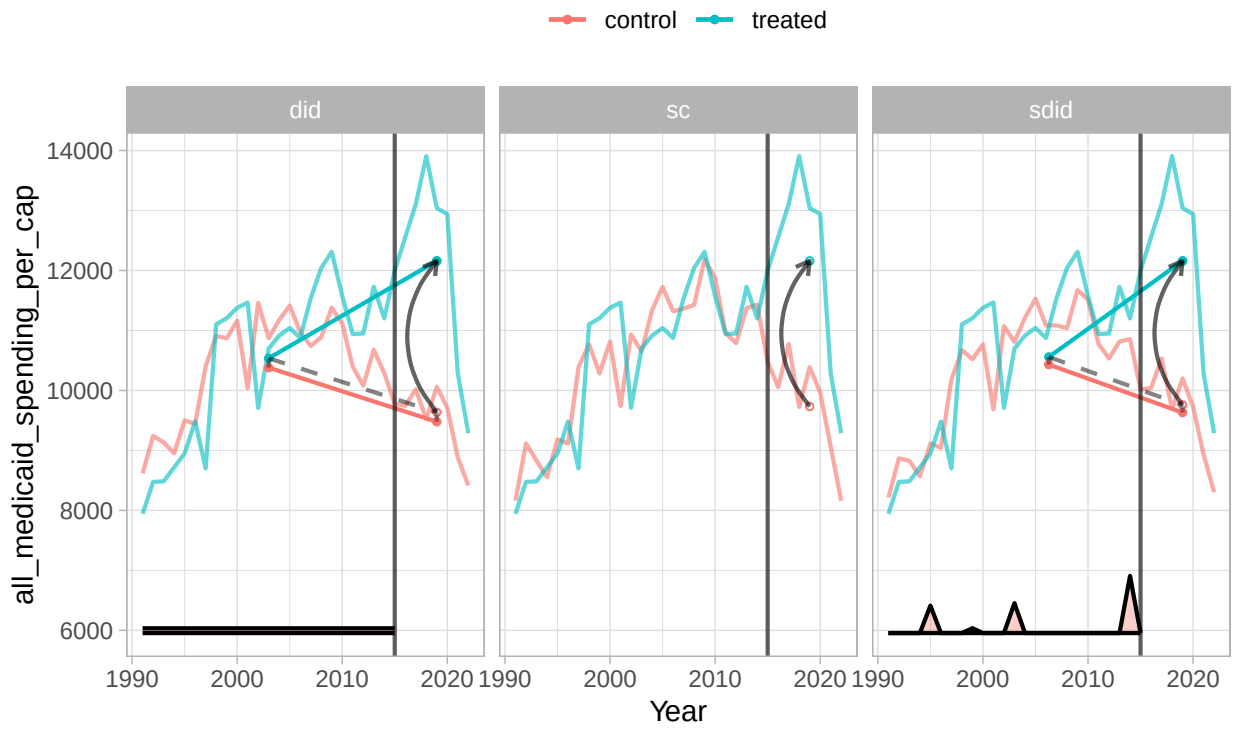



```
## Plot saved for cohort year 2012
```

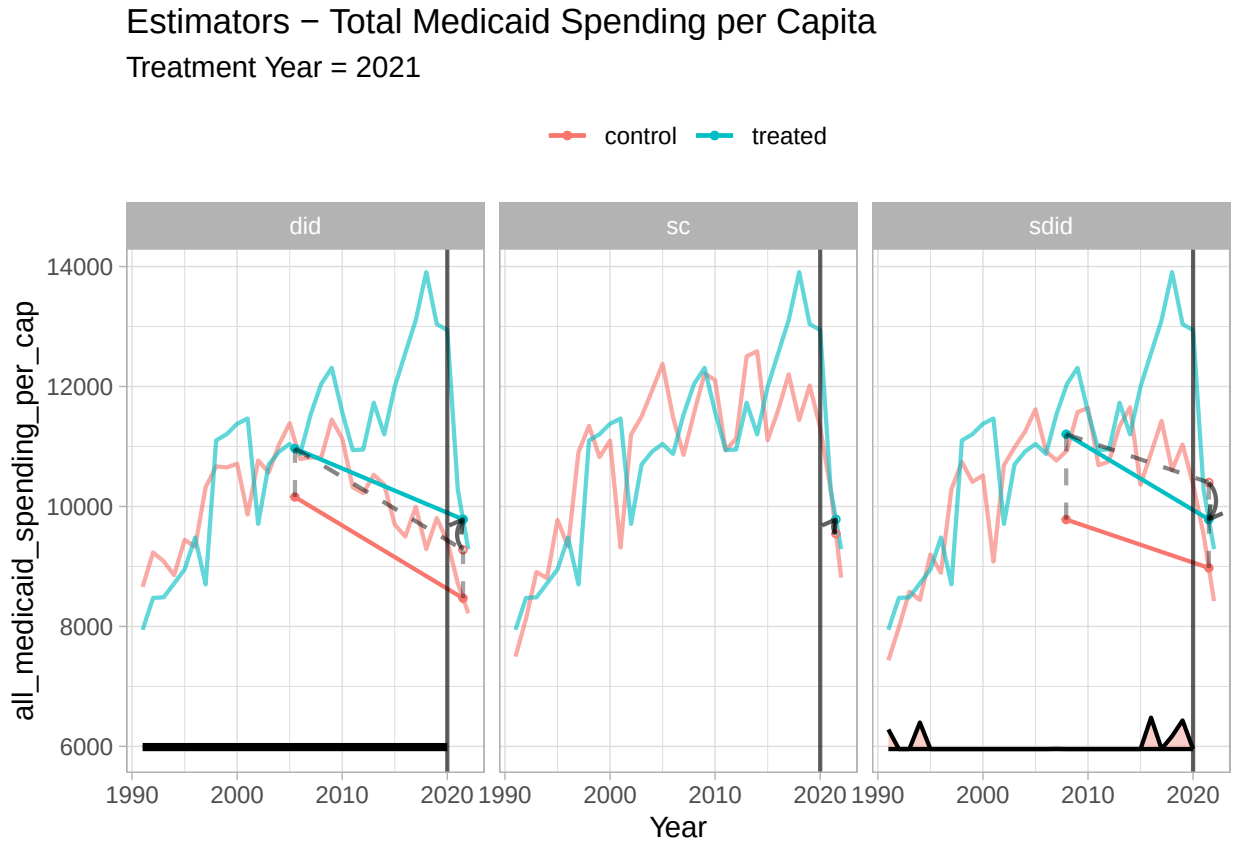



```
## Plot saved for cohort year 2014
```


```
## Plot saved for cohort year 2016
```




```
## Plot saved for cohort year 2021
```

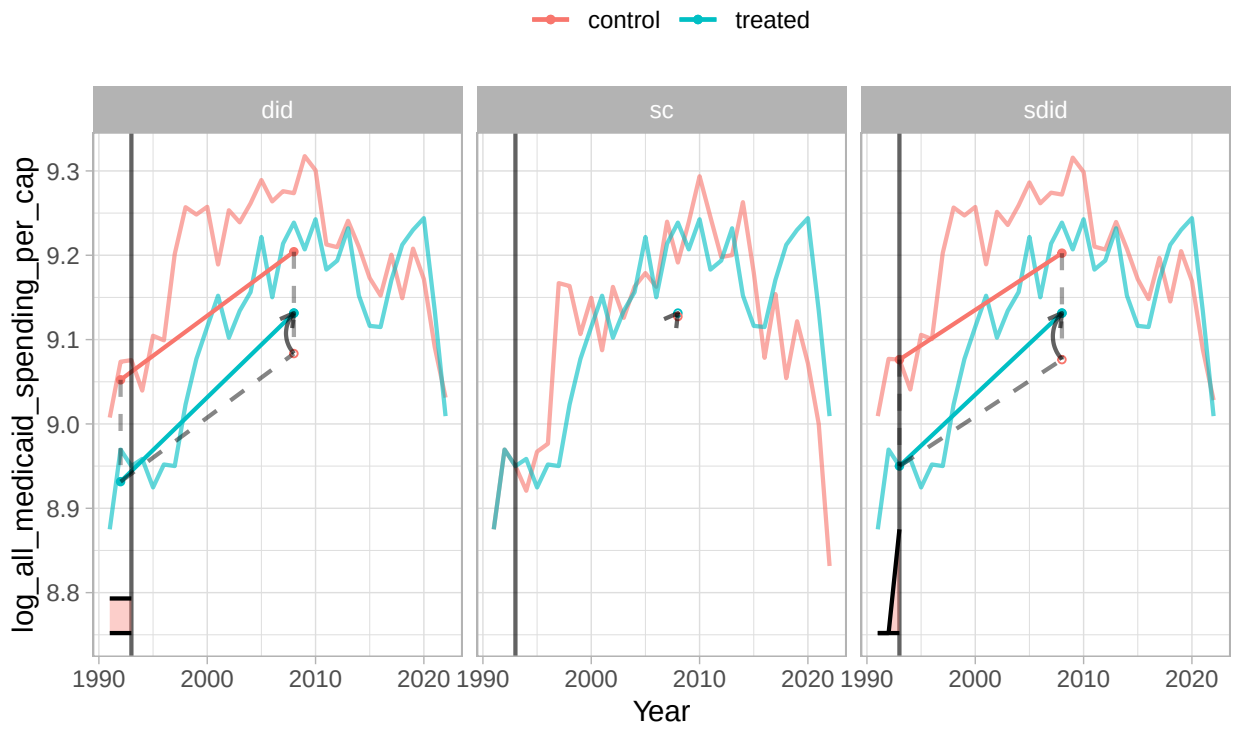


Log(Total Medicaid Spending per Capita)

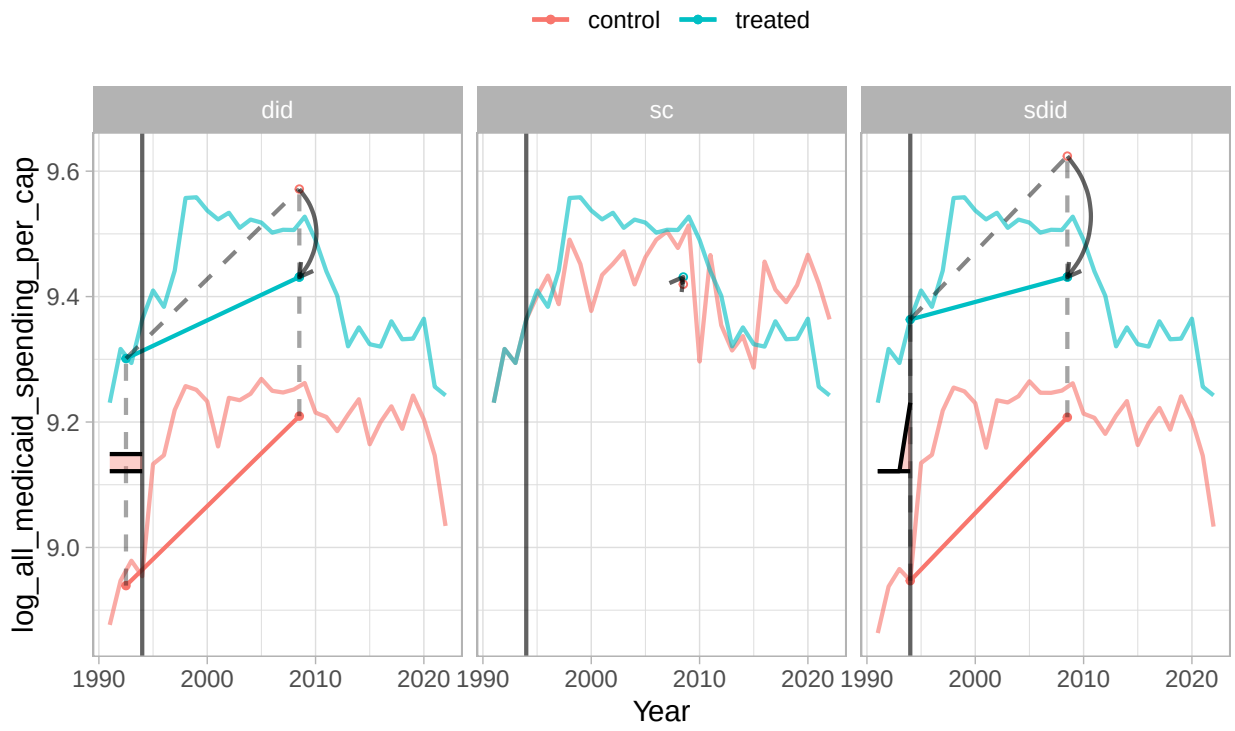
```
run_synthdid(data, "log_all_medicaid_spending_per_cap", cohorts)
```

```
## Error in cohort year 1992: dim(X) must have a positive length
```

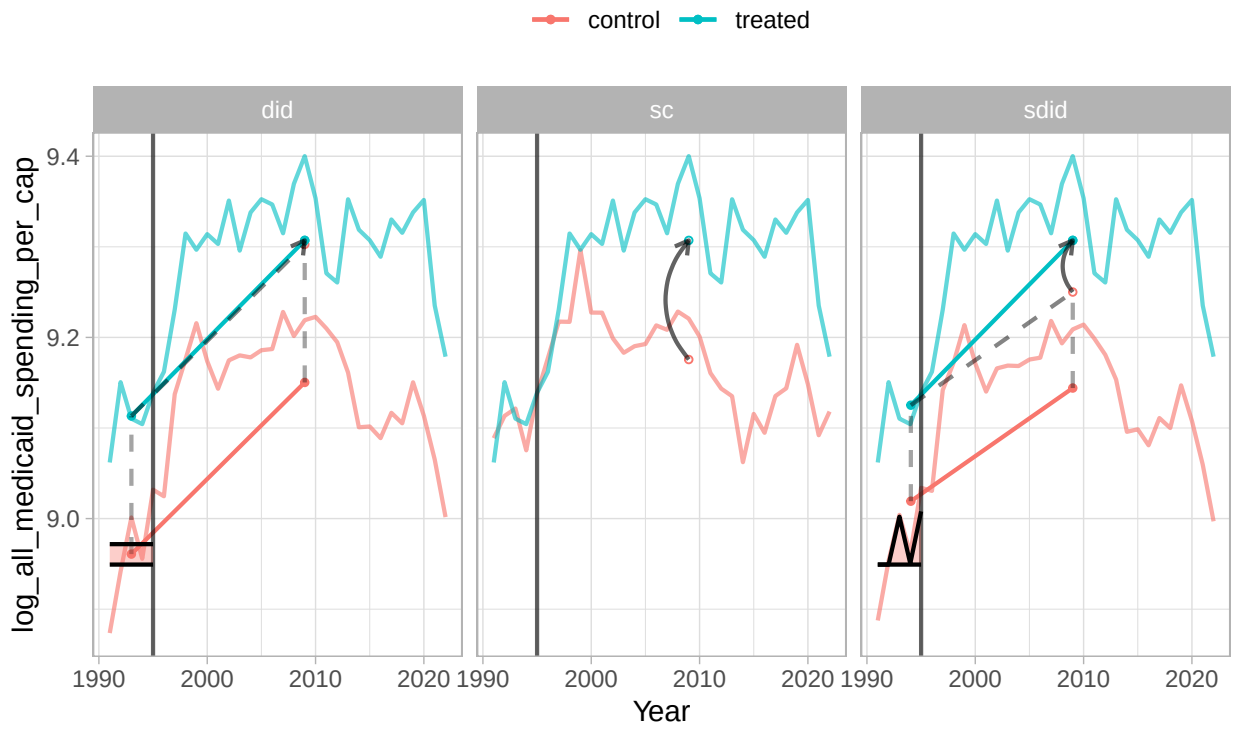
```
## Plot saved for cohort year 1994
```



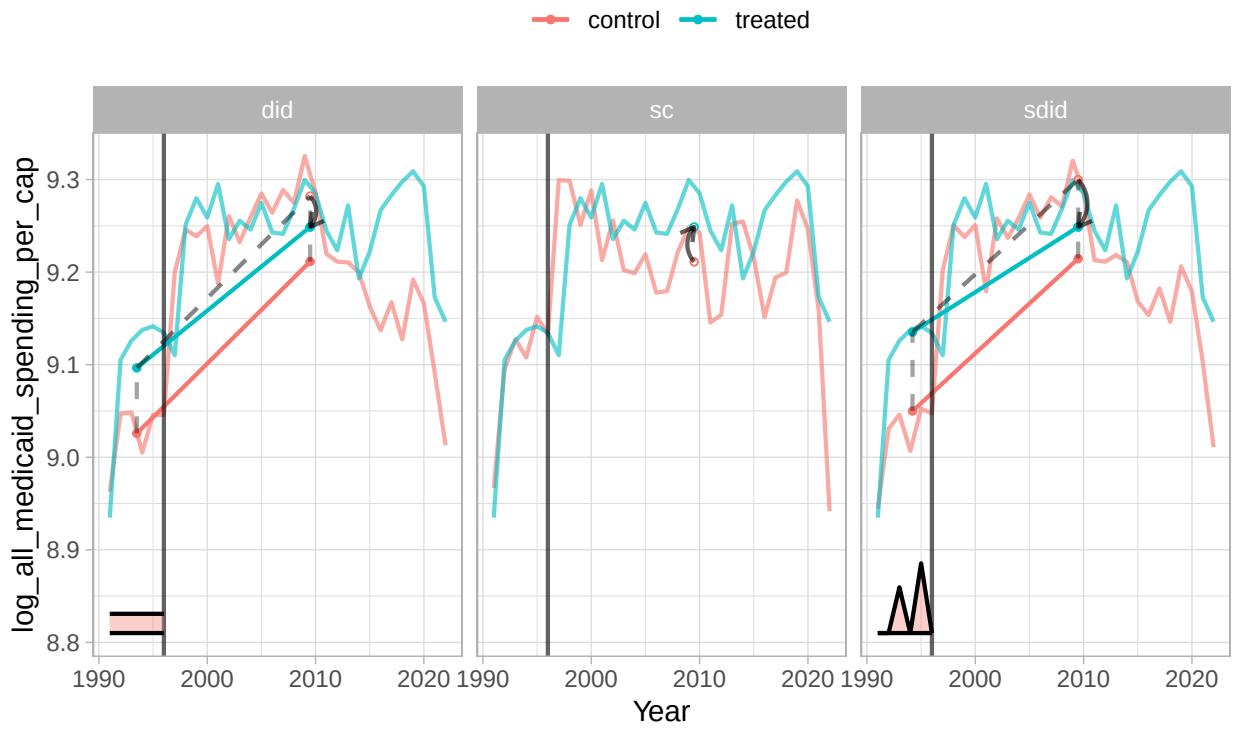

```
## Plot saved for cohort year 1995
```



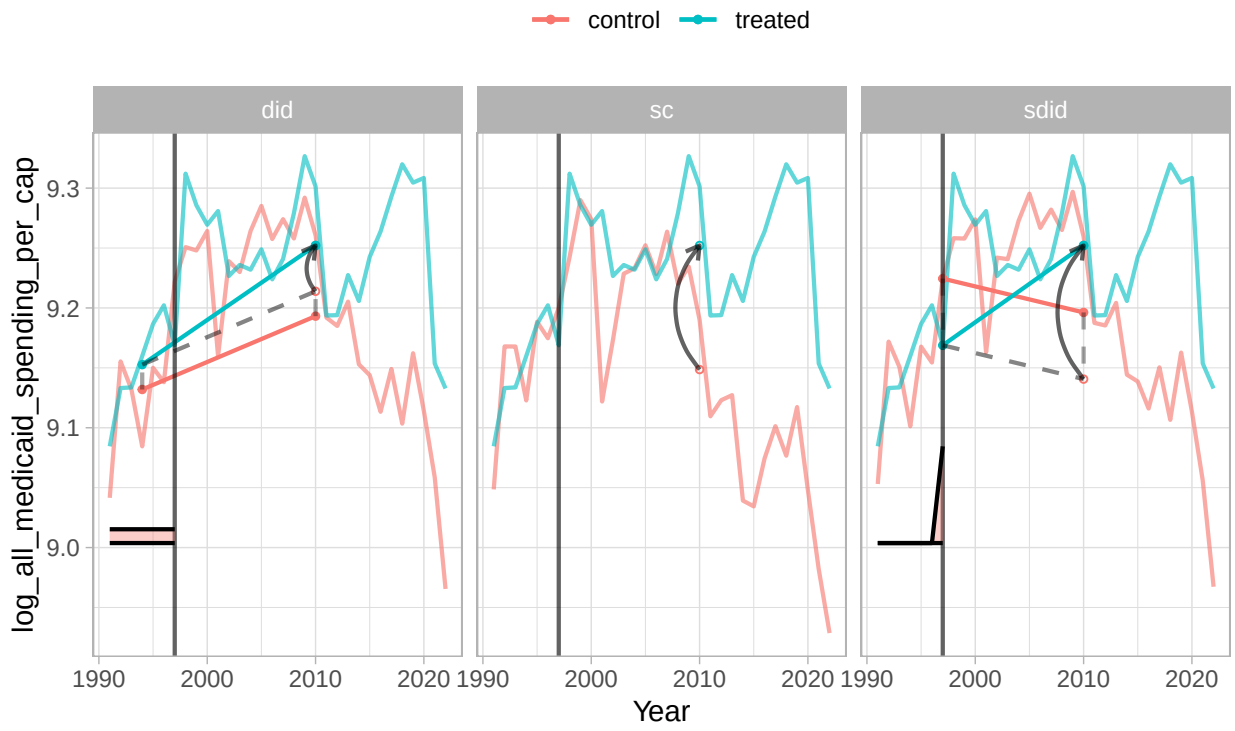

```
## Plot saved for cohort year 1996
```



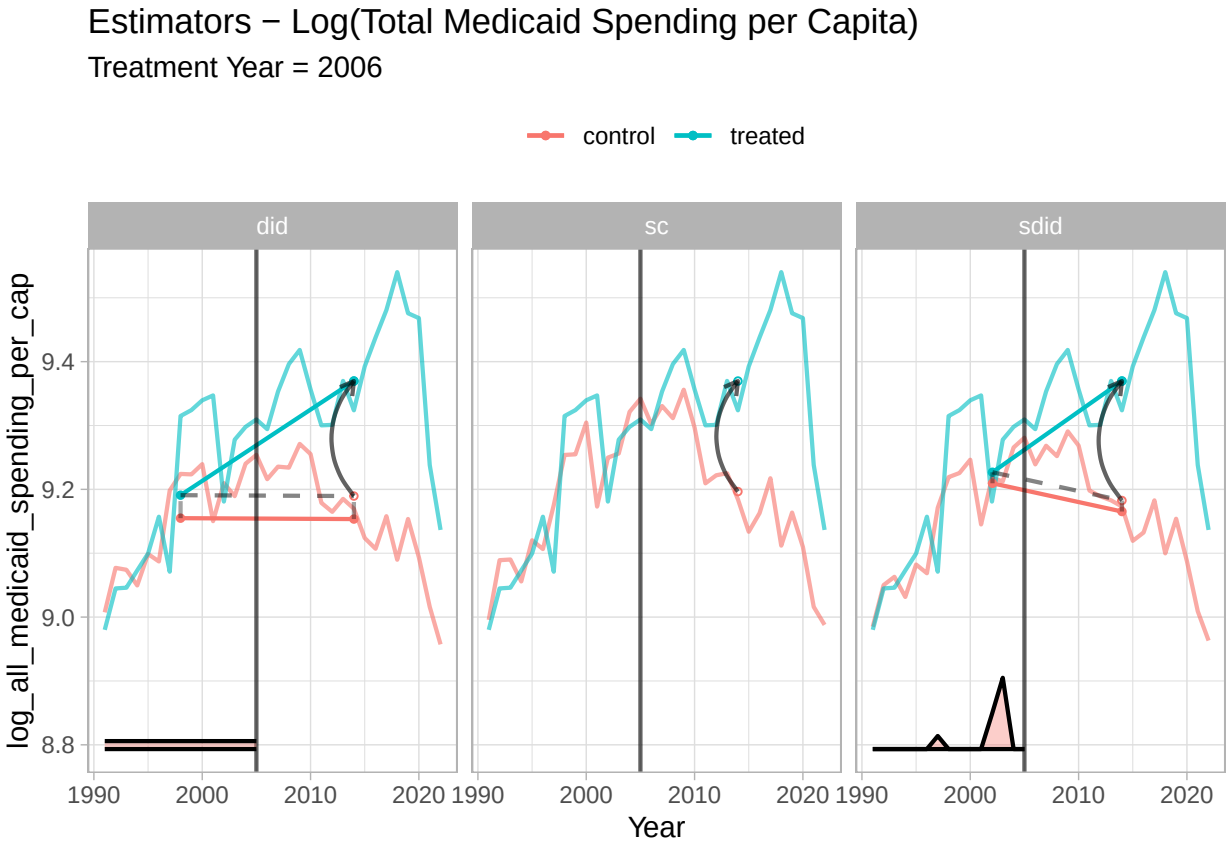

```
## Plot saved for cohort year 1997
```



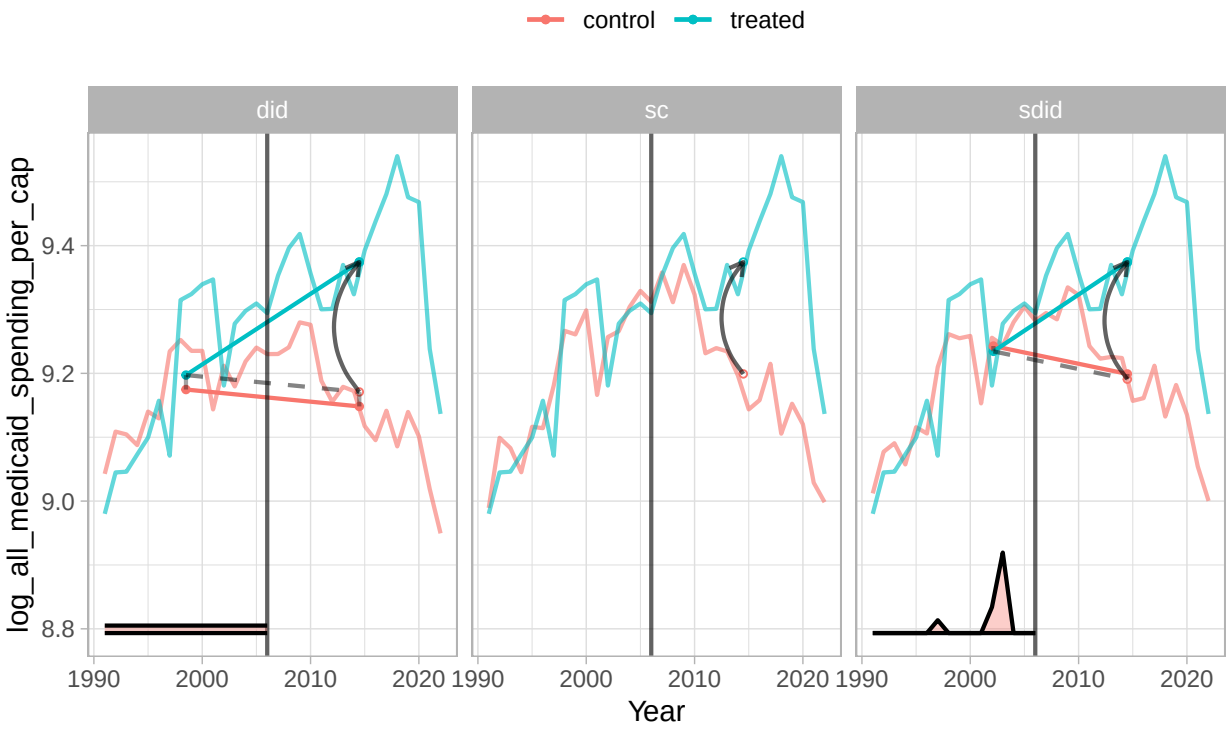

```
## Plot saved for cohort year 1998
```



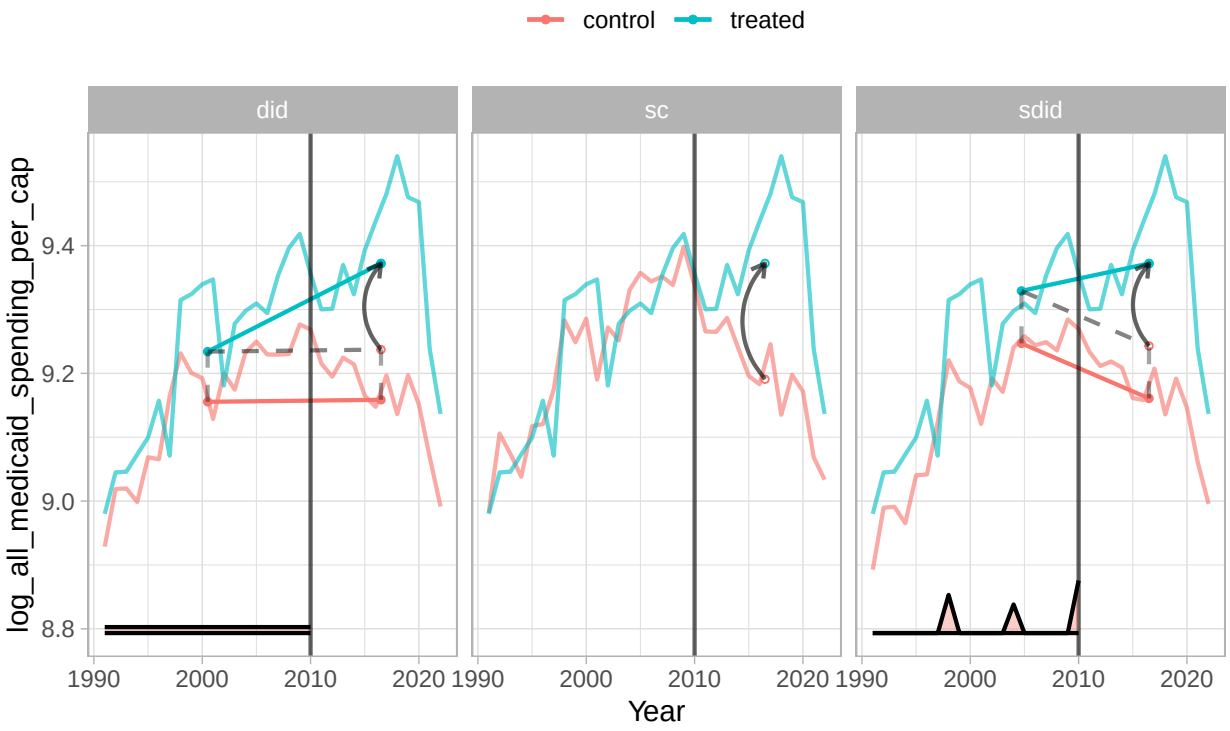

```
## Plot saved for cohort year 2006
```



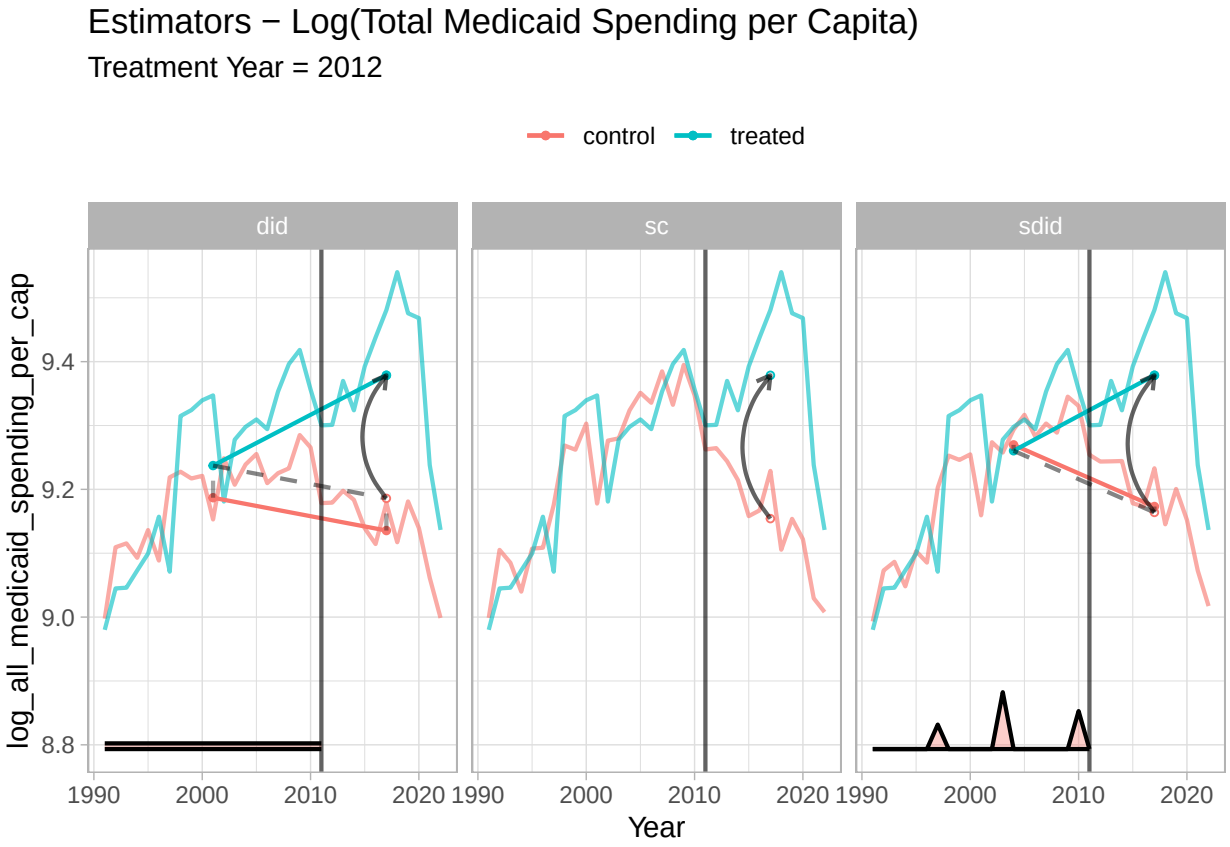

```
## Plot saved for cohort year 2007
```



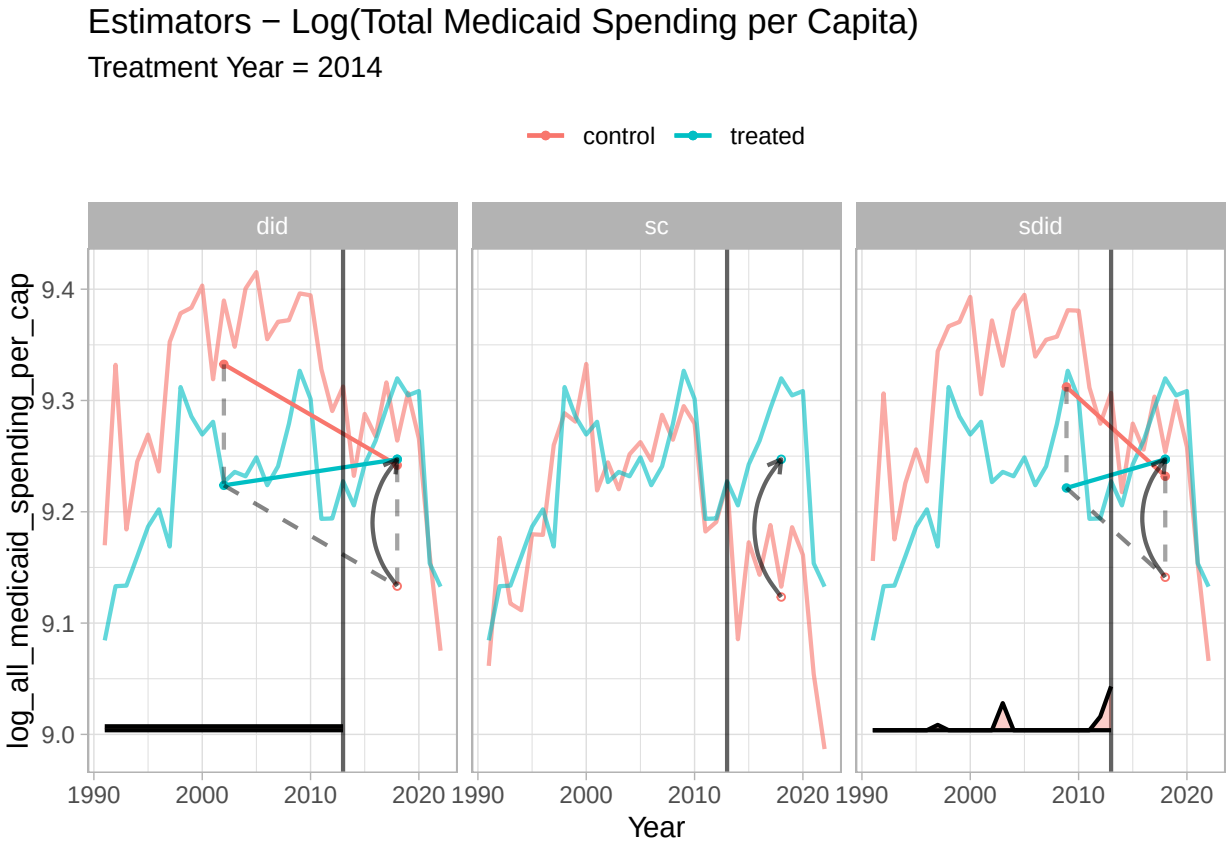

```
## Plot saved for cohort year 2011
```



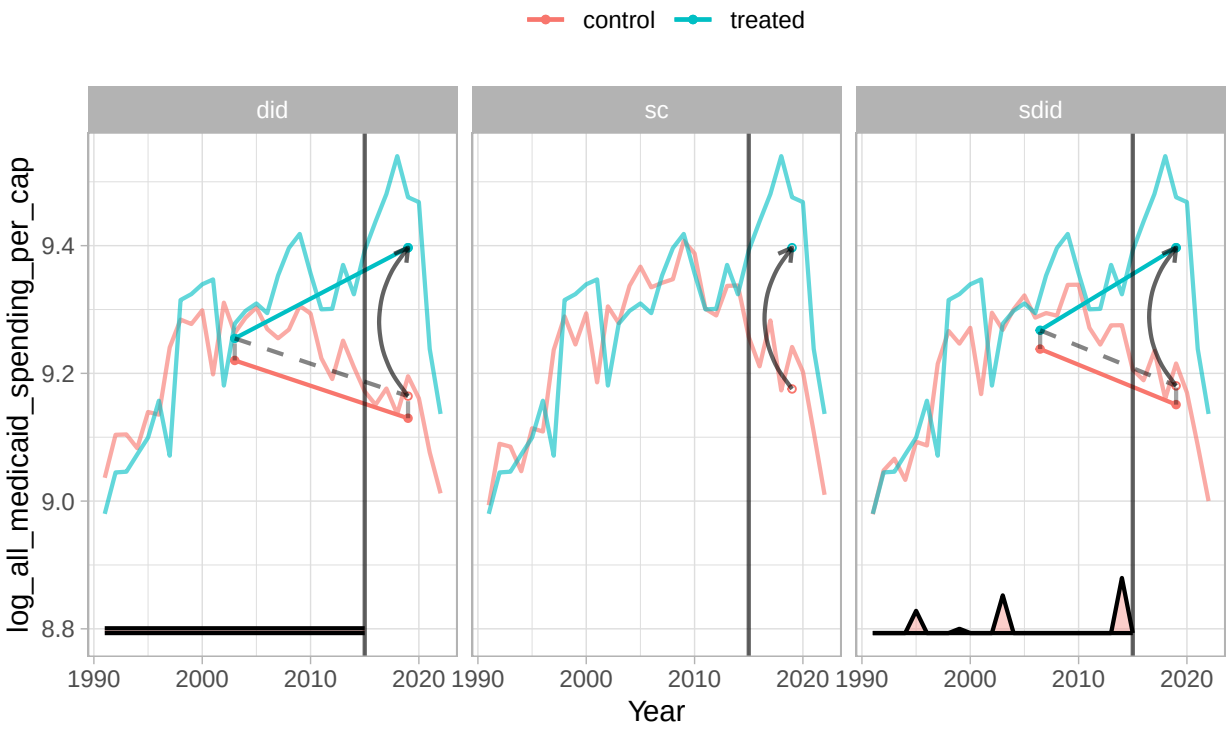

```
## Plot saved for cohort year 2012
```



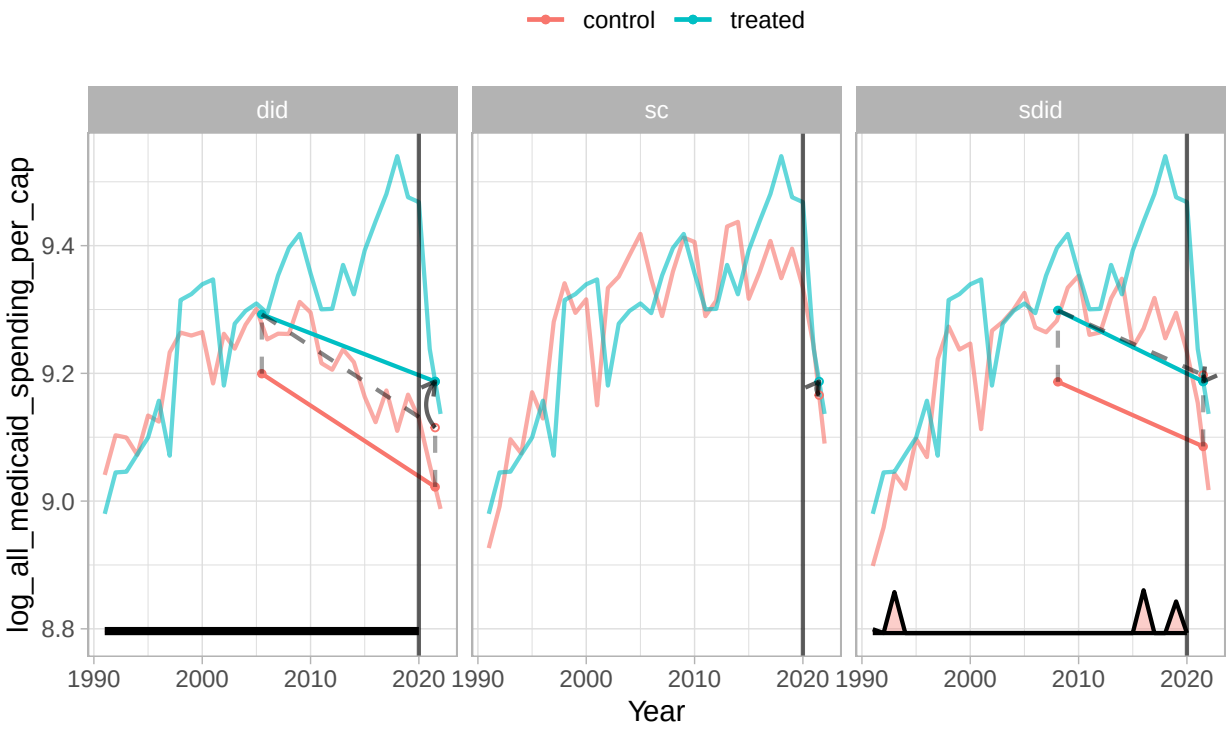

```
## Plot saved for cohort year 2014
```




```
## Plot saved for cohort year 2016
```




```
## Plot saved for cohort year 2021
```



TWFE Treatment Effect by Cohort (cohorts based off 5-10 year intervals)

```
# Define 5 and 10 year interval cohorts based on treatment years
data_cohorts <- data %>%
  mutate(cohort_5_interval = case_when(
    treatment_year >= 1991 & treatment_year <= 1995 ~ "1991-1995",
    treatment_year >= 1996 & treatment_year <= 2000 ~ "1996-2000",
    treatment_year >= 2001 & treatment_year <= 2005 ~ "2001-2005",
    treatment_year >= 2006 & treatment_year <= 2010 ~ "2006-2010",
    treatment_year >= 2011 & treatment_year <= 2015 ~ "2011-2015",
    treatment_year >= 2016 & treatment_year <= 2022 ~ "2016-2022",
    TRUE ~ "never-treated"
  )) %>%
  mutate(cohort_10_interval = case_when(
    treatment_year >= 1991 & treatment_year <= 2000 ~ "1991-2000",
    treatment_year >= 2001 & treatment_year <= 2010 ~ "2001-2010",
    treatment_year >= 2011 & treatment_year <= 2022 ~ "2011-2022",
    TRUE ~ "never-treated"
  ))
```

```
# Vector of relative time variables
keepvars <- c(paste0("`center_time_-", 10:2, "`"),
              paste0("center_time_", 0:10))
```

```r
prep_es_10 <- function(mod){
  mod_2 <- tibble(
    estimate = mod$coefficients,
    term1 = rownames(mod$coefficients),
    se = mod$se,
    pval = mod$pval
  )

  # Filter coefficients for relative time leads and lags
  es <- mod_2 %>%
    filter(term1 %in% keepvars) %>%
    mutate(t = c(-10:-2, 0:10)) %>%
    select(t, estimate, se, pval)

  # Add a row for t = -1 (reference period)
  es <- rbind(es, c(-1, 0, 0, 0))

  # Add confidence intervals
  es <- es %>%
    mutate(lower = estimate - 1.96 * se,
           upper = estimate + 1.96 * se)

  return(es)
}

prep_es_trend_10 <- function(mod){
  mod_2 <- tibble(
    estimate = mod$coefficients,
    term1 = rownames(mod$coefficients),
    se = mod$se,
    pval = mod$pval
  )

  es <- mod_2 %>%
    filter(term1 %in% keepvars) %>%
    mutate(t = c(0:10)) %>%
    select(t, estimate, se, pval)
  es <- rbind(es, c(-1, 0, 0, 0))
  es <- es %>%
    mutate(lower = estimate - 1.96 * se,
           upper = estimate + 1.96 * se,
           star = ifelse(pval <= 0.001, "***",
                         ifelse(pval<=0.01,"**",
                                ifelse(pval<=0.05, "*", ""))),
           coef_star = paste0(round(estimate,2), star, sep =""))
  return(es)
}

#' Plot 10-year event study plot with data filtered by interval cohort
#' @param data panel data
#' @param outcome outcome variable (numeric)
#' @param outcome_name name of the outcome variable (character)
#' @param cohort_var name of the interval variable (character)
```

```r
#' @param cohort_interval range of years (cohort) we want to filter for (character)
did_graph_trend_10 <- function(data, outcome, outcome_name, cohort_var, cohort_interval){
  p <- list()

  cohort_data <- data %>%
    filter(!!sym(cohort_var) == cohort_interval)

  # Normal event study
  formula_str1 <- paste0(
    outcome, " ~ ",
    "`center_time_-10` + `center_time_-9` + ",
    "`center_time_-8` + `center_time_-7` + `center_time_-6` + `center_time_-5` + ",
    "`center_time_-4` + `center_time_-3` + `center_time_-2` + `center_time_0` + ",
    "`center_time_1` + `center_time_2` + `center_time_3` + `center_time_4` + ",
    "`center_time_5` + `center_time_6` + `center_time_7` + `center_time_8` + ",
    "`center_time_9` + `center_time_10` | ",
    "as.factor(state) + as.character(year) ",
    sep = ""
  )
  formula1 <- as.formula(formula_str1)
  mod_d <- lfe::felm(formula1,
                     data = subset(cohort_data, center_time >= -10 & center_time <= 10),
                     exactDOF = TRUE)
  es_d <- prep_es_10(mod_d)

  es_d <- es_d %>%
    mutate(sig = ifelse((lower > 0 | upper < 0), 1, 0))

  # Pre-trend line
  pre_trend <- es_d %>% filter(t < 0)
  pre_trend_fit <- lm(estimate ~ t, data = pre_trend)
  predicted_df <- data.frame(
    t = seq(-10, 10, by = 0.1),
    estimate = predict(pre_trend_fit, newdata = data.frame(t = seq(-10, 10, by = 0.1)))
  )

  ## estimate difference
  formula_str2 <- paste0(
    outcome, " ~ ",
    "`center_time_0` + ",
    "`center_time_1` + `center_time_2` + `center_time_3` + `center_time_4` + ",
    "`center_time_5` + `center_time_6` + `center_time_7` + `center_time_8` + ",
    "`center_time_9` + `center_time_10` + ",
    "center_time | ",
    "as.factor(state) + as.character(year) ",
    sep = ""
  )
  formula2 <- as.formula(formula_str2)
  mod_d <- lfe::felm(formula2,
                     data = subset(cohort_data, center_time >= -10 & center_time <= 10),
                     exactDOF = TRUE)
  es_d2 <- prep_es_trend_10(mod_d)
```

```r
  es_d2 <- es_d2 %>%
    mutate(sig = ifelse((lower > 0 | upper < 0), 1, 0))

  # Event study plot
  p1 <- ggplot(data = es_d, aes(x = t, y = estimate, group = 1))+
    geom_point(size= 3, color = "gray26")+
    geom_errorbar(aes(ymin = lower, ymax = upper),
                  width=.1,
                  linewidth = 0.8,
                  color = "gray26") +
    geom_hline(yintercept = 0,
               linetype = "dashed",
               linewidth = 1,
               color = "firebrick4")+
    scale_x_continuous(breaks = round(seq(-10, 10, by = 1),1),
                       limits = c(-10.5, 10.5))+
    # scale_y_continuous(breaks = round(seq(-0.2, 0.2, by = 0.1), 1),
    #                    limits = c(-0.2, 0.2))+
    labs(x = "Relative Time", y = "Estimate",
         title = paste0(outcome_name, " - ", cohort_interval))+
    theme(plot.title = element_text(hjust = 0.5, size = 15),
          legend.position = "none")+
    geom_text(data = es_d2 %>% filter(t >= 0 & t <= 10),
              aes(x = t, y = 0.5*max(es_d2$upper), label = coef_star, angle = 45),
              color = "blue", size = 4, vjust = -0.3)+
    geom_line(data = predicted_df, aes(x = t, y = estimate),
              color = "blue", linetype = "dashed", linewidth = 1.2)
  # annotate("text", x = 8, y = text_pos, label = comb_est, size = 5)

  p2 <- ggplot(data = es_d2, aes(x = t, y = estimate)) +
    geom_point(size = 3, color = "black") +  # Black dots for estimates
    geom_errorbar(aes(ymin = lower, ymax = upper),
                  width = 0.2,
                  linewidth = 0.8,
                  color = "black") +  # Error bars
    geom_line(color = "black", linewidth = 1) +  # Line connecting estimates
    geom_hline(yintercept = 0, linetype = "dashed", color = "firebrick4", linewidth = 1) +  # Reference
    scale_x_continuous(breaks = seq(-20, 20, by = 1)) +
    # scale_y_continuous(breaks = seq(-0.5,1,0.1),
    #                    limits = c(-0.5, 1.1))+
    labs(x = "Relative Time",
         y = "Estimate",
         title = paste0(outcome_name, " - ", cohort_interval)) +
    theme(plot.title = element_text(hjust = 0.5, size = 15),
          legend.position="none")
  p2

  p[[1]] <- p1
  p[[2]] <- p2
  return(p)
}
```

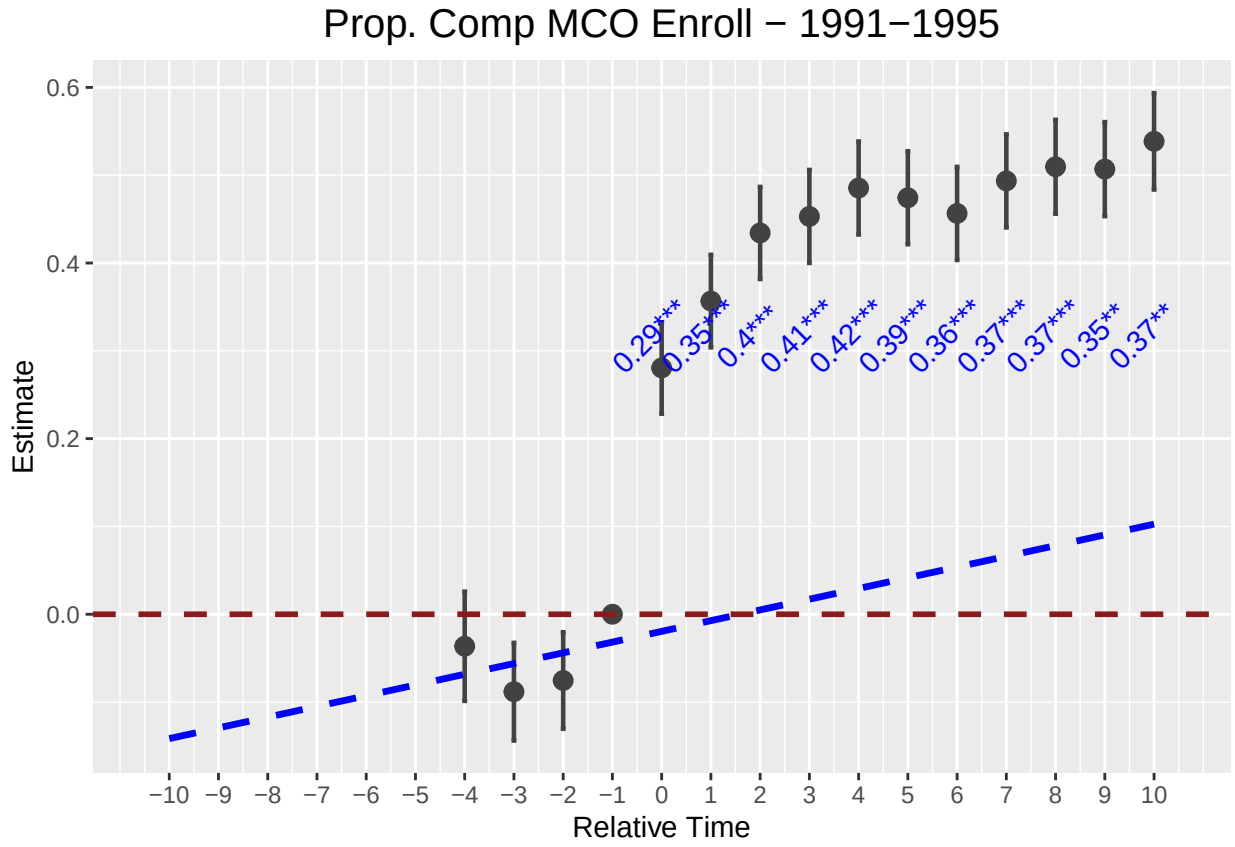
```r
# Define treatment cohorts
assign_cohort_5 <- function(treatment_year) {
  case_when(
    is.na(treatment_year) ~ "never-treated",
    treatment_year >= 1996 & treatment_year <= 2000 ~ "1996-2000",
    treatment_year >= 1991 & treatment_year <= 1995 ~ "1991-1995",
    treatment_year >= 2006 & treatment_year <= 2010 ~ "2006-2010",
    treatment_year >= 2016 & treatment_year <= 2022 ~ "2016-2022",
    treatment_year >= 2011 & treatment_year <= 2015 ~ "2011-2015",
    TRUE ~ "never-treated"  # Default fallback if none of the above match
  )
}

# Construct stacked data
# For each treatment year, construct control without treatment within 10/20 years
collect_10 <- list()
collect_20 <- list()

for(i in unique(data_cohorts$treatment_year[data_cohorts$treatment_year != Inf])){

  # Get treated state
  treated_state <- data_cohorts %>%
    filter(treatment_year == i) %>%
    mutate(treatment_year_group = i) %>%
    mutate(cohort_5_interval = assign_cohort_5(treatment_year_group)) %>%
    select(state, year, center_time, treated, total_medicaid,
           log_all_medicaid_spending_per_cap, all_medicaid_spending_per_cap,
           pct_in_comp_mco, treatment_year, treatment_year_group, cohort_5_interval)

  max_year <- max(treated_state$year)

  # Get control states within a 10 year (pre and post) from treatment year
  control_state_10 <- data_cohorts %>%
    filter(treatment_year > i + 10,
           year <= i + 10, year >= i - 10) %>%
    mutate(treatment_year_group = i,
           center_time = 0,
           treated = 0,
           cohort_5_interval = assign_cohort_5(treatment_year_group)) %>%
    select(state, year, center_time, treated, total_medicaid,
           log_all_medicaid_spending_per_cap, all_medicaid_spending_per_cap,
           pct_in_comp_mco, treatment_year, treatment_year_group, cohort_5_interval)

  control_state_20 <- data_cohorts %>%
    filter(treatment_year > i + 20,
           year <= i + 20, year >= i - 20) %>%
    mutate(treatment_year_group = i,
           center_time = 0,
           treated = 0,
           cohort_5_interval = assign_cohort_5(treatment_year_group)) %>%
    select(state, year, center_time, treated, total_medicaid,
           log_all_medicaid_spending_per_cap, all_medicaid_spending_per_cap,
           pct_in_comp_mco, treatment_year, treatment_year_group, cohort_5_interval)
```

```
  collect_10[[length(collect_10)+1]] <- rbind(treated_state, control_state_10)
  collect_20[[length(collect_20)+1]] <- rbind(treated_state, control_state_20)
}

# Collapse into one dataframe
dat_10 <- do.call(rbind, collect_10)
dat_20 <- do.call(rbind, collect_20)

dat_10 <- dat_10 %>% dummy_cols(select_columns = "center_time")
dat_20 <- dat_20 %>% dummy_cols(select_columns = "center_time")

# DID - 10 period
dat <- dat_10
dat <- dat %>%
  mutate(log_total_spend = log(total_medicaid))
```

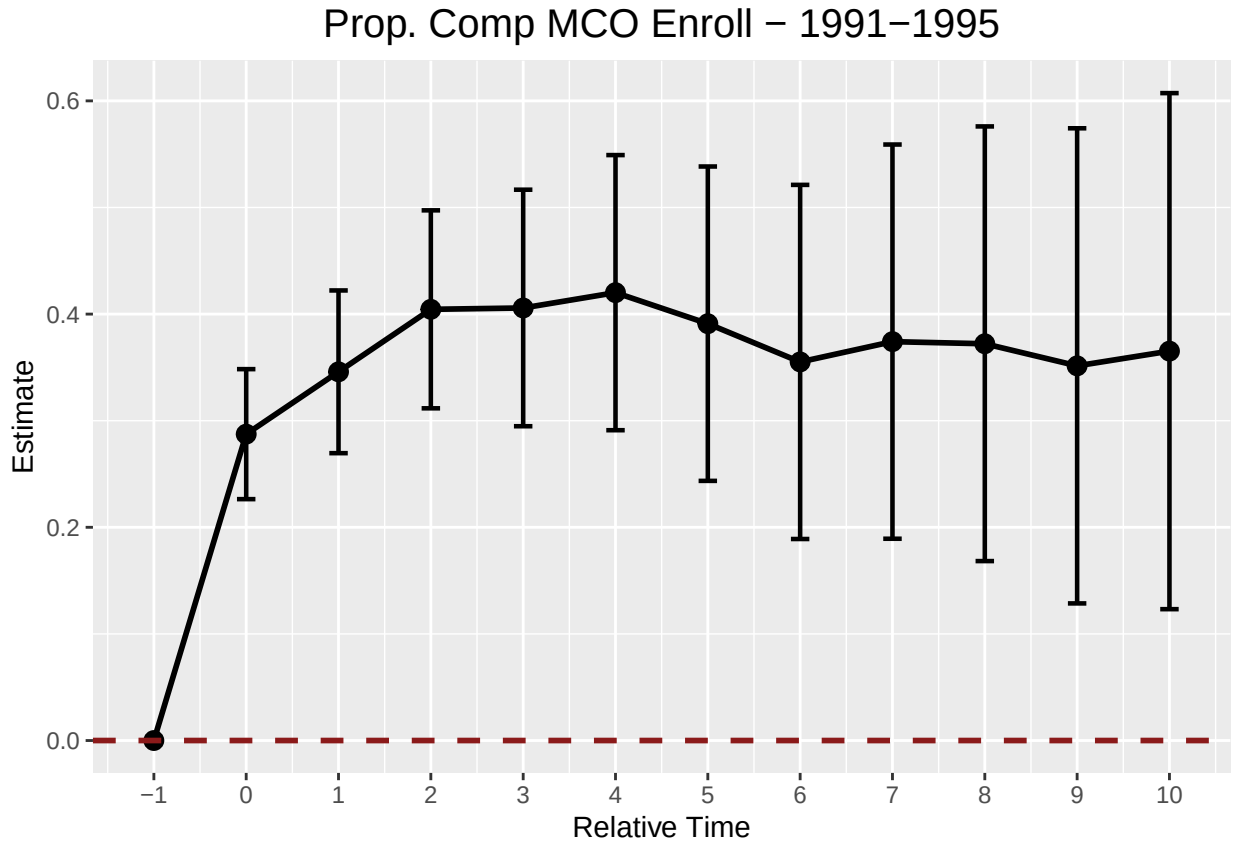
Plots for 5 year interval cohorts

```
cohorts <- c("1991-1995", "1996-2000", "2006-2010", "2011-2015", "2016-2022")

outcomes <- list(
  pct_in_comp_mco = "Prop. Comp MCO Enroll",
  log_total_spend = "Log(Total Medicaid Spending)",
  log_all_medicaid_spending_per_cap = "Log(Total Medicaid Spending per Capita)"
)


# Initialize an empty list to store the plots
plot_list <- list()

# Loop through each outcome
for (outcome_name in names(outcomes)) {
  outcome_label <- outcomes[[outcome_name]]

  # Loop through each cohort
  for (cohort in cohorts) {
    # Generate the plot
    p <- did_graph_trend_10(dat, outcome_name, outcome_label, "cohort_5_interval", cohort)

    # Store the plot in the list
    plot_list[[paste(outcome_name, cohort, sep = "_")]] <- p
  }

  outcome_plots <- plot_list[grep(outcome_name, names(plot_list))]
  print(outcome_plots)
}
```

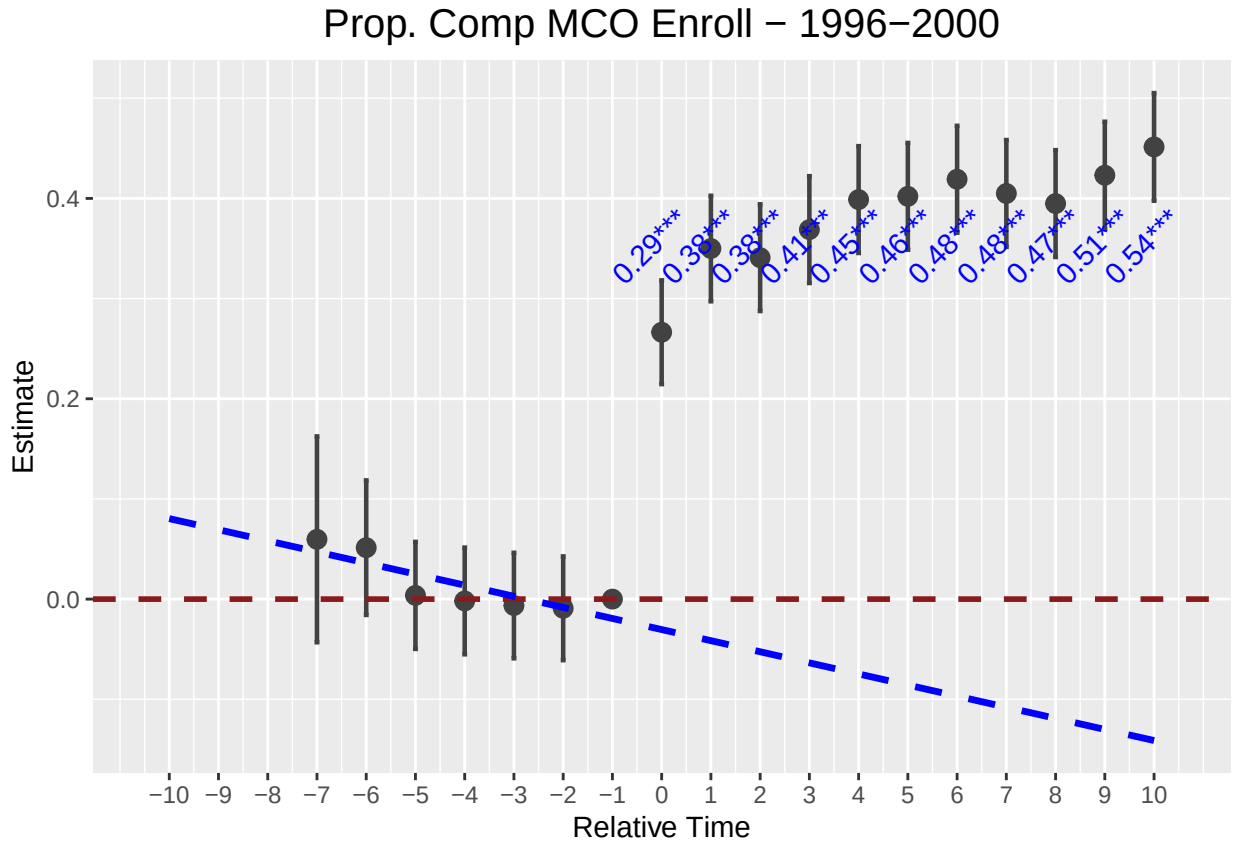
```
## $`pct_in_comp_mco_1991-1995`
## $`pct_in_comp_mco_1991-1995`[[1]]
```



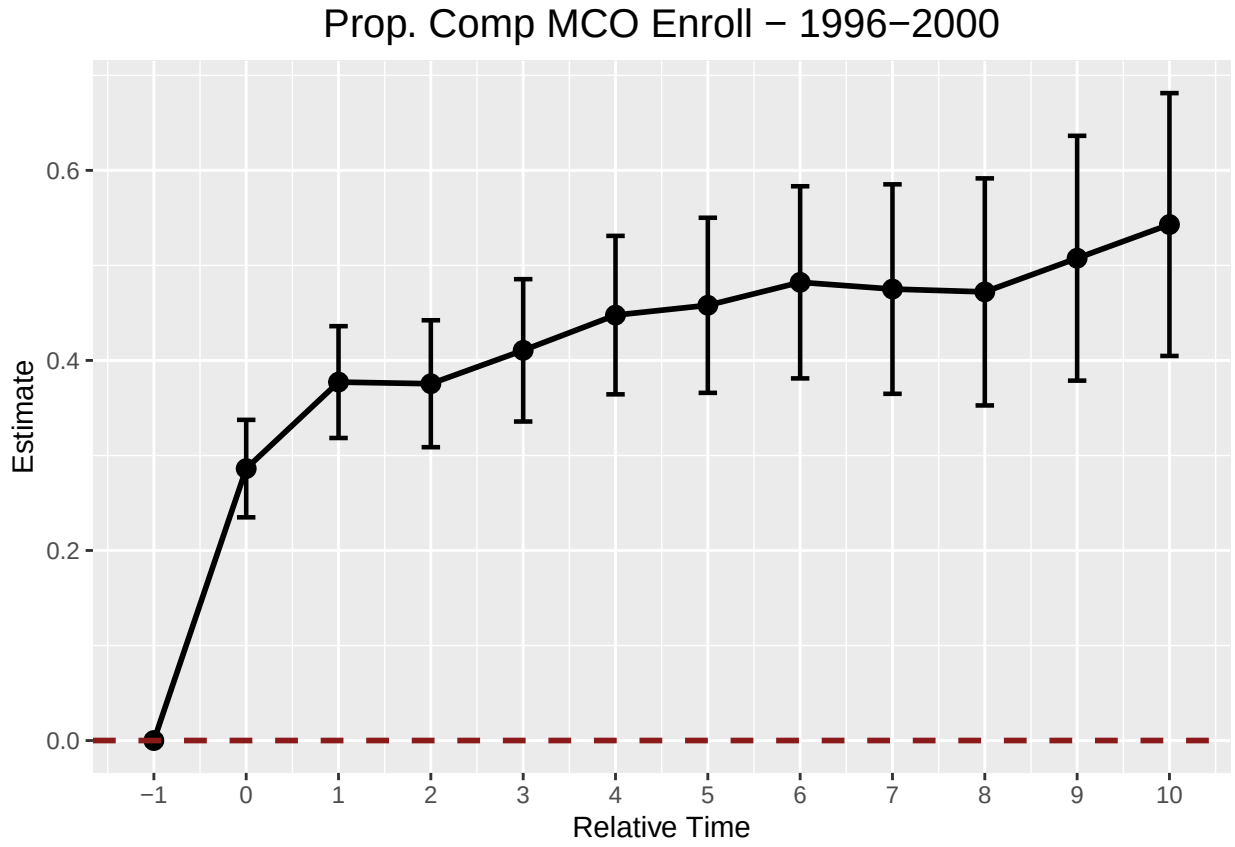

```
##
## $`pct_in_comp_mco_1991-1995`[[2]]
```



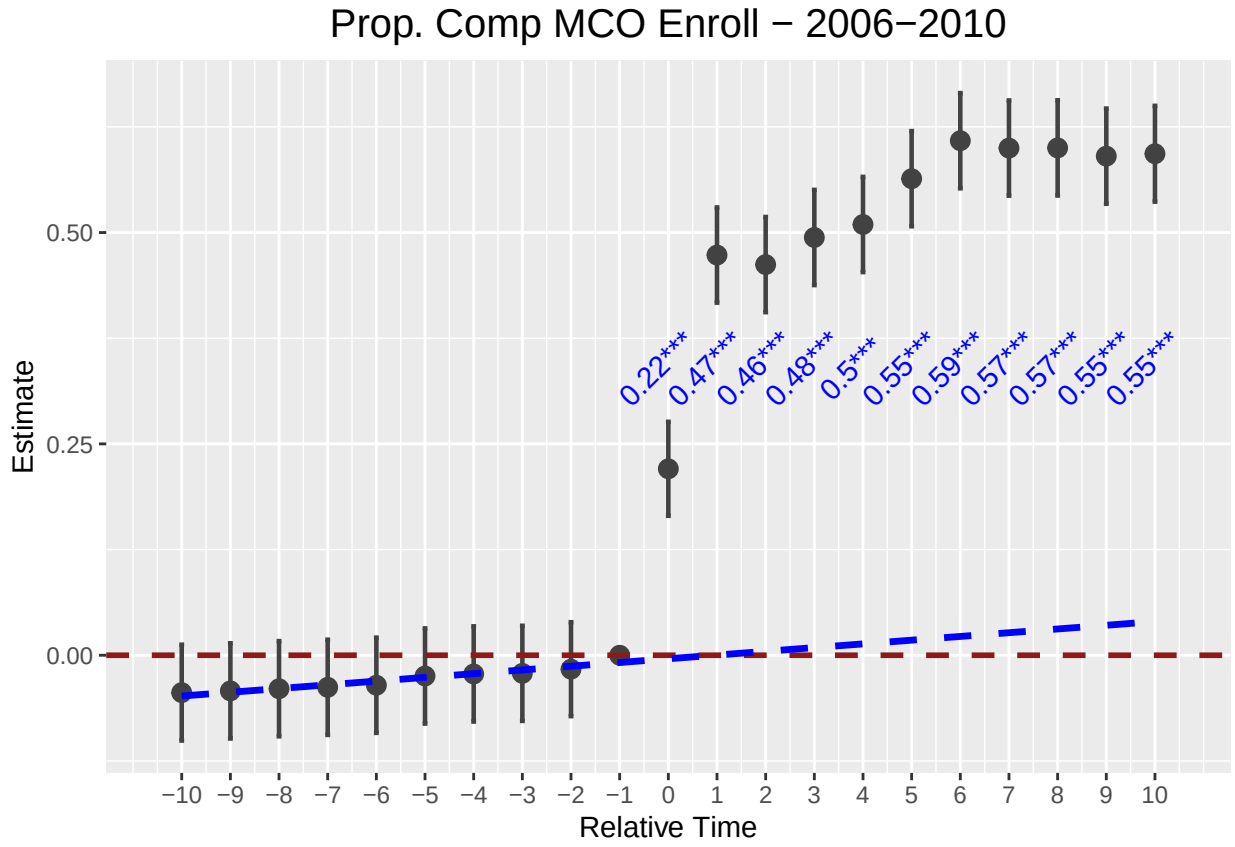

```
##
##
## $`pct_in_comp_mco_1996-2000`
## $`pct_in_comp_mco_1996-2000`[[1]]
```



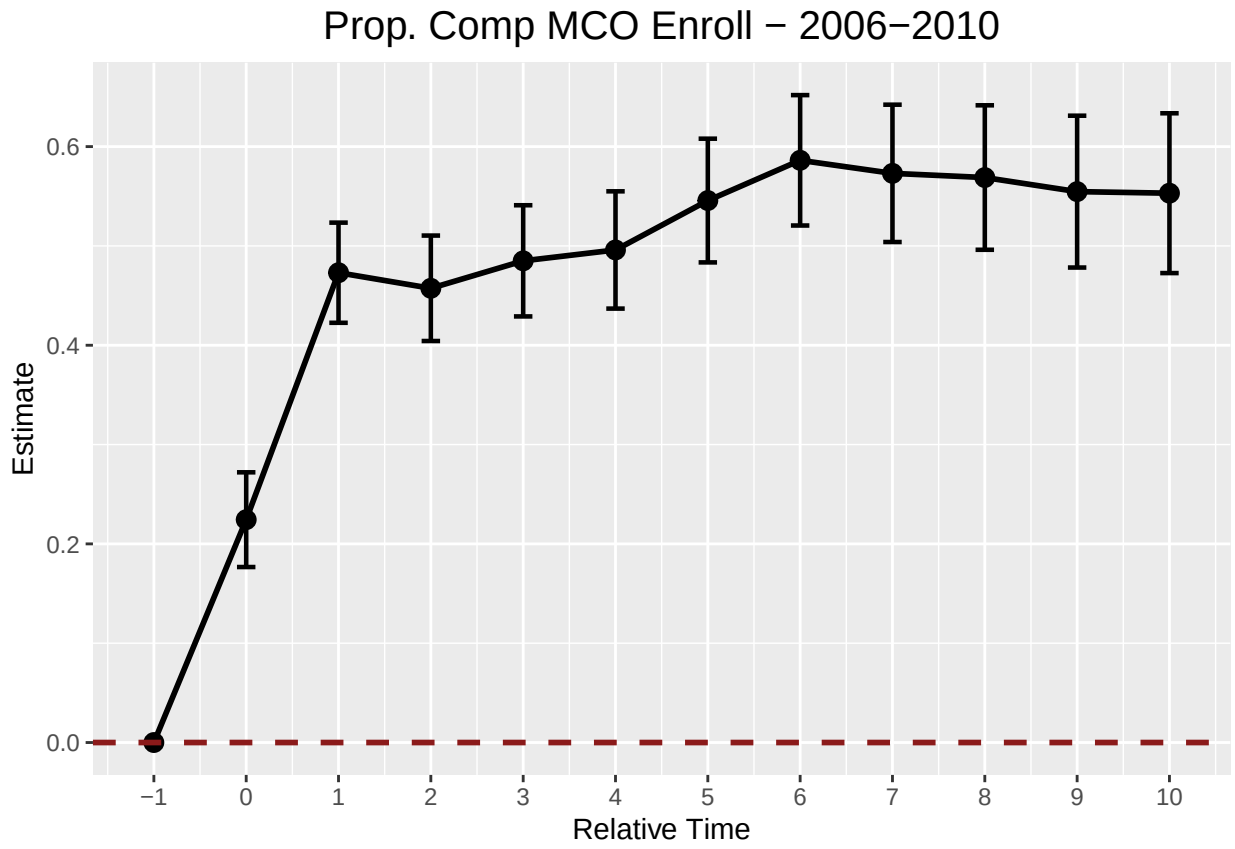

```
##
## $`pct_in_comp_mco_1996-2000`[[2]]
```



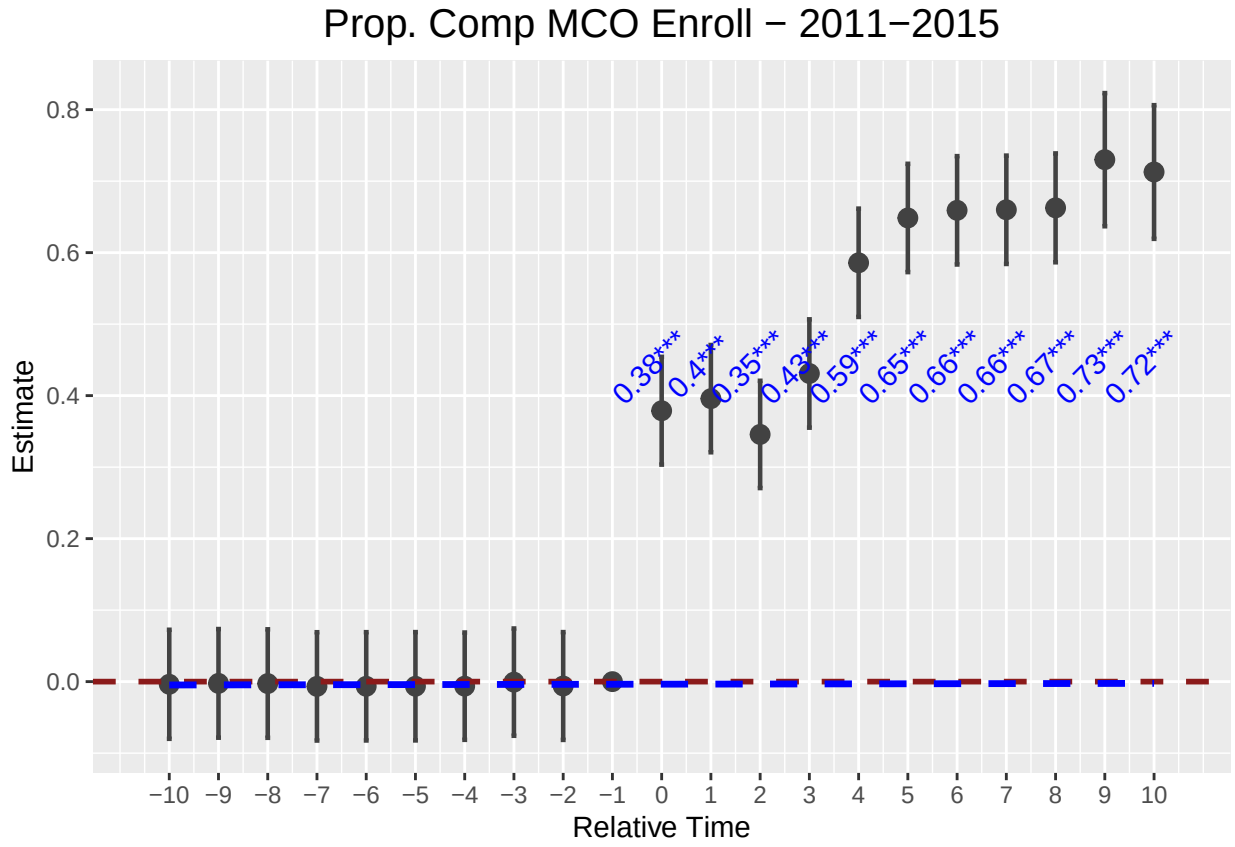

```
##
##
## $`pct_in_comp_mco_2006-2010`
## $`pct_in_comp_mco_2006-2010`[[1]]
```



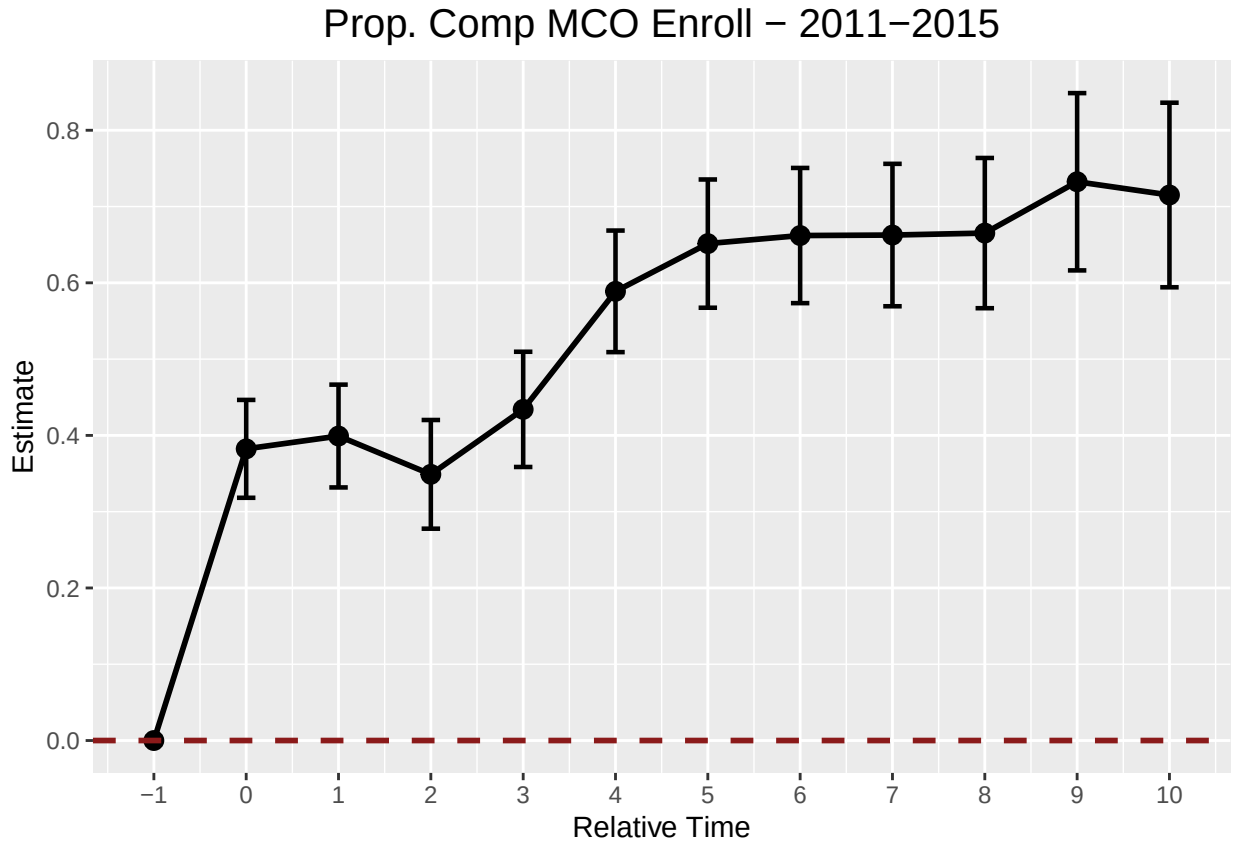

```
##
## $`pct_in_comp_mco_2006-2010`[[2]]
```



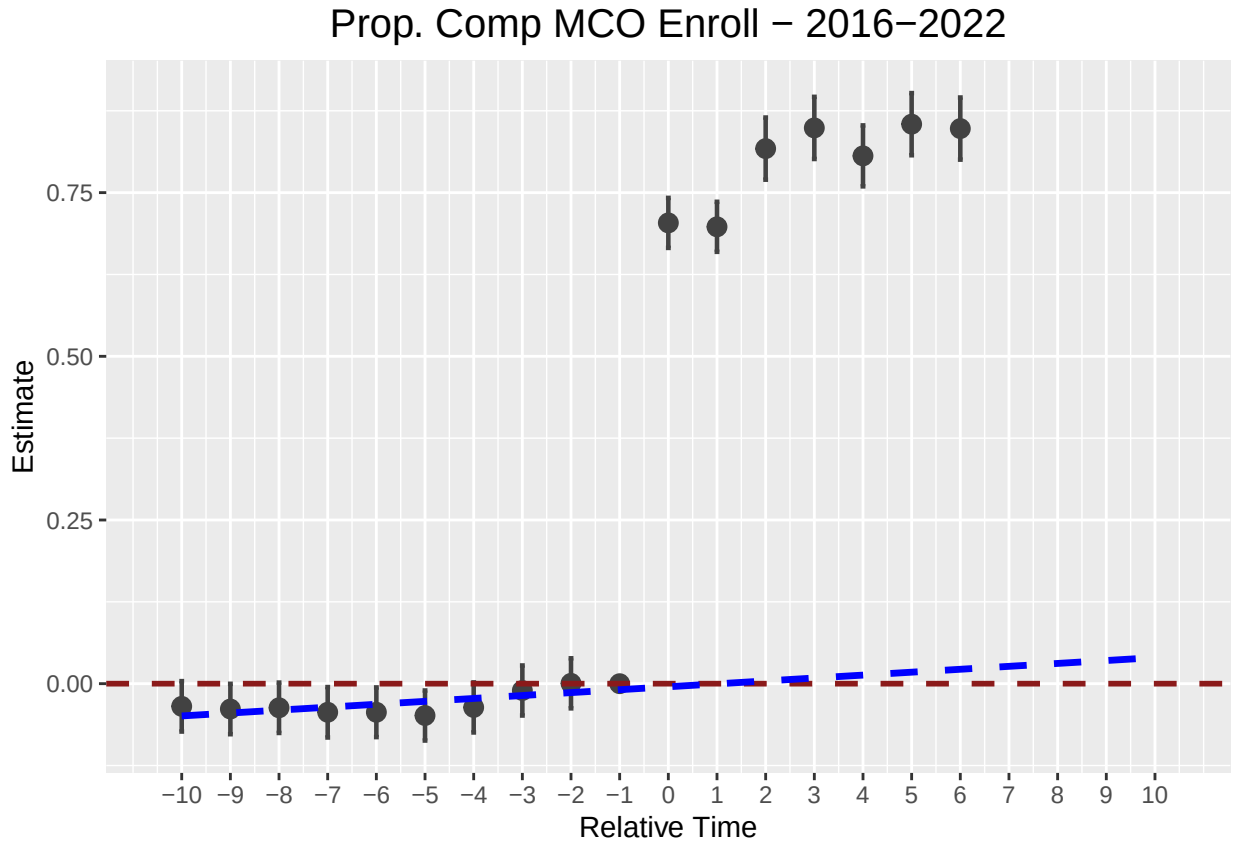

```
##
##
## $`pct_in_comp_mco_2011-2015`
## $`pct_in_comp_mco_2011-2015`[[1]]
```



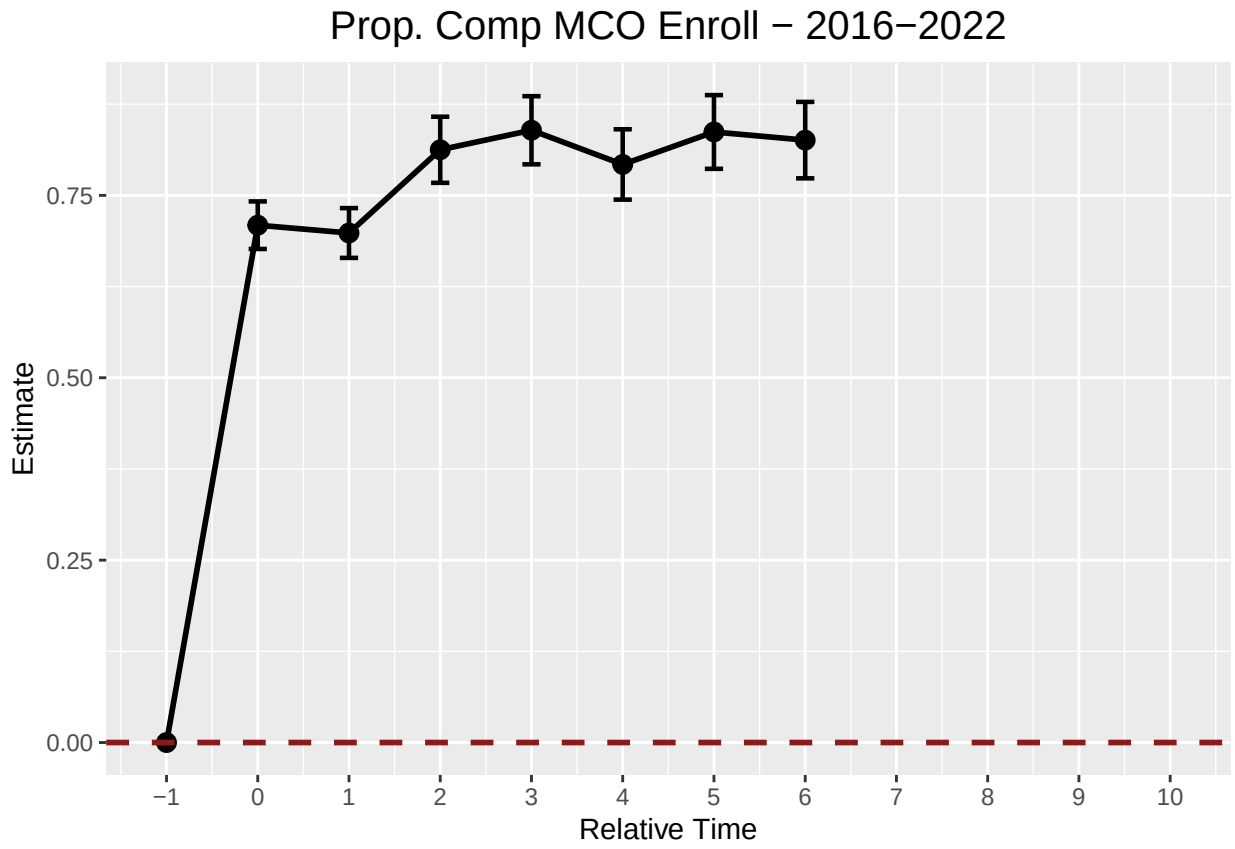

```
##
## $`pct_in_comp_mco_2011-2015`[[2]]
```



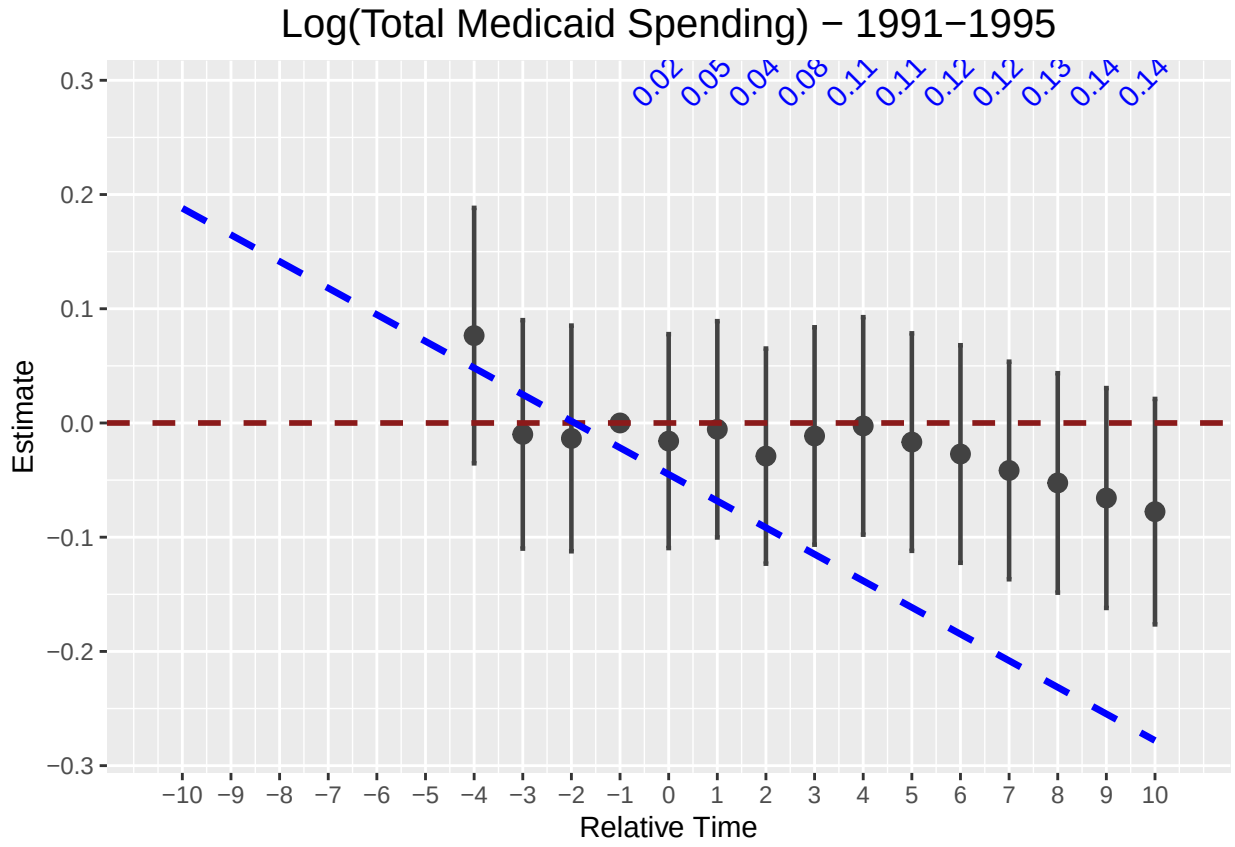

```
##
##
## $`pct_in_comp_mco_2016-2022`
## $`pct_in_comp_mco_2016-2022`[[1]]
```



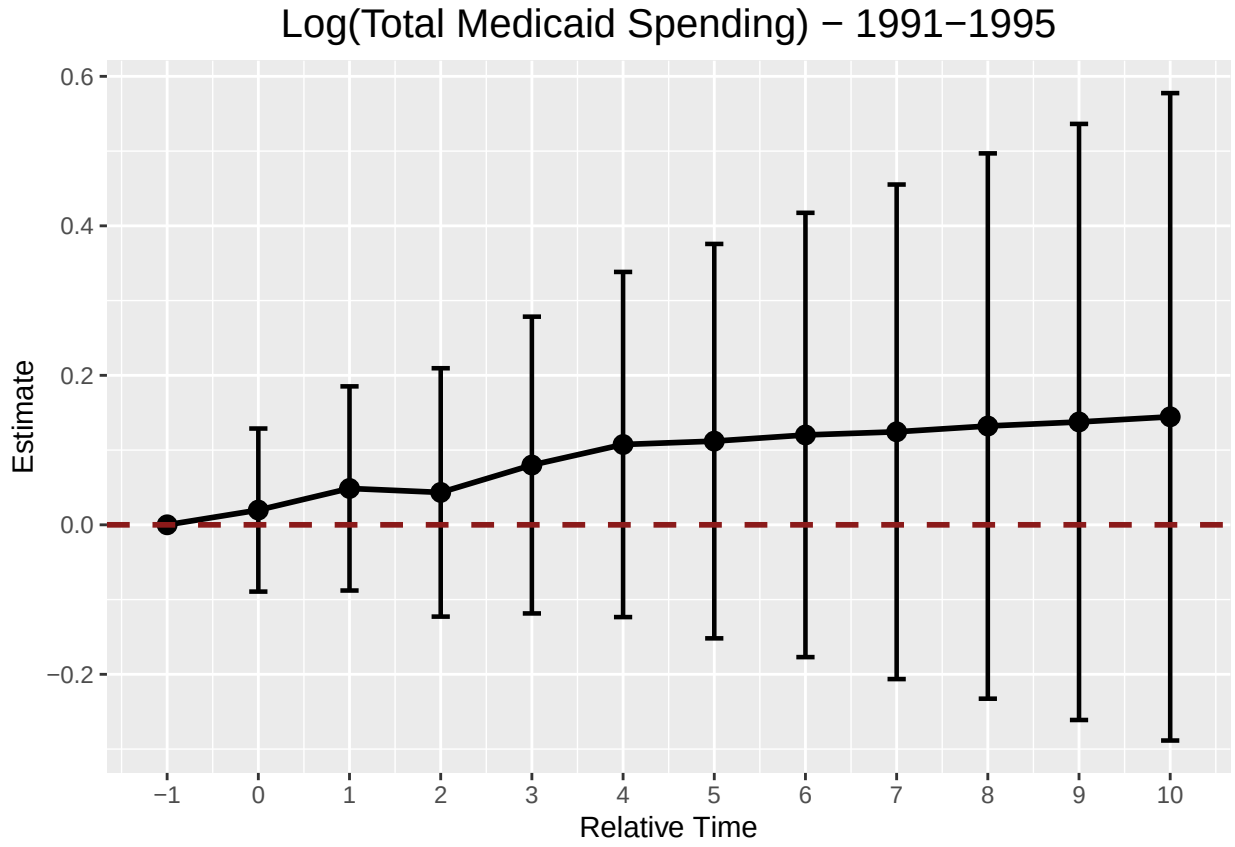

```
##
## $`pct_in_comp_mco_2016-2022`[[2]]
```



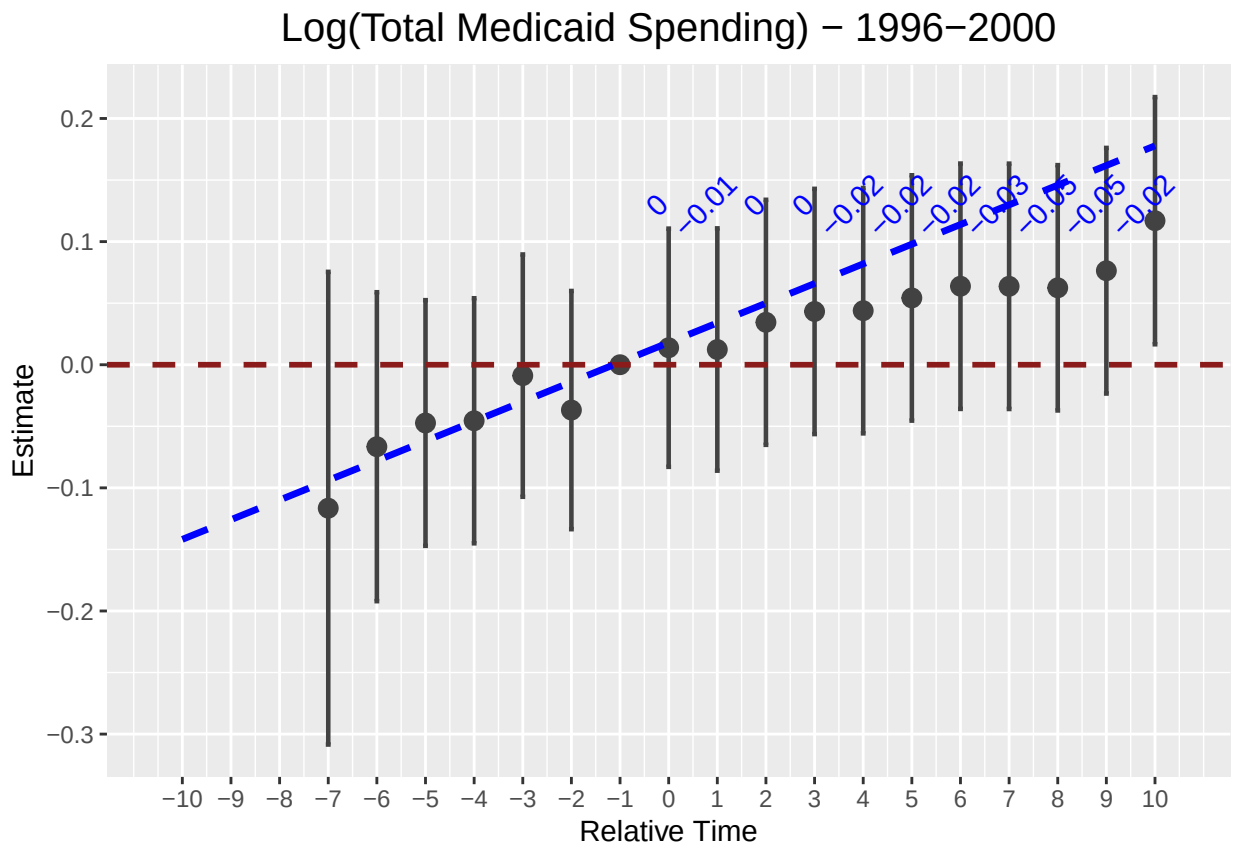

```
##
##
## $`log_total_spend_1991-1995`
## $`log_total_spend_1991-1995`[[1]]
```



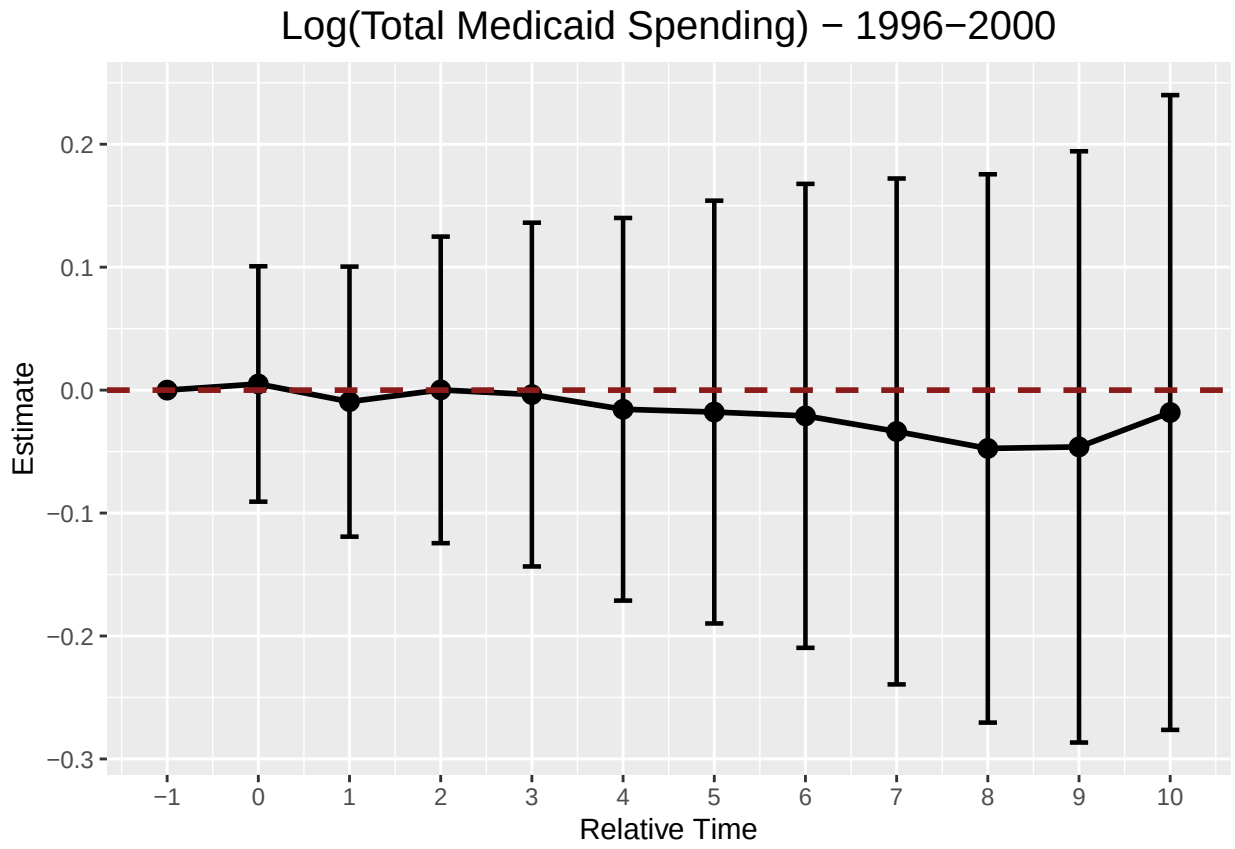

```
##
## $`log_total_spend_1991-1995`[[2]]
```



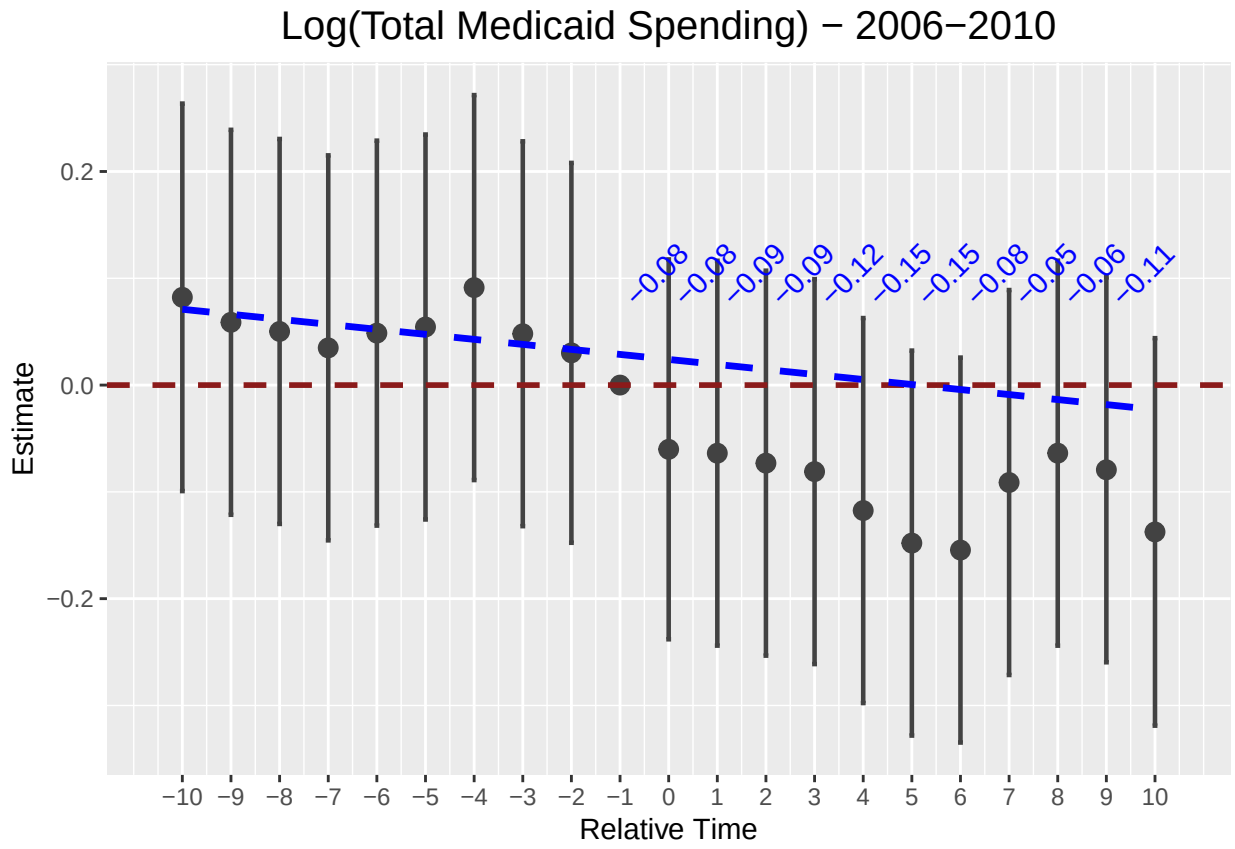

```
##
##
## $`log_total_spend_1996-2000`
## $`log_total_spend_1996-2000`[[1]]
```



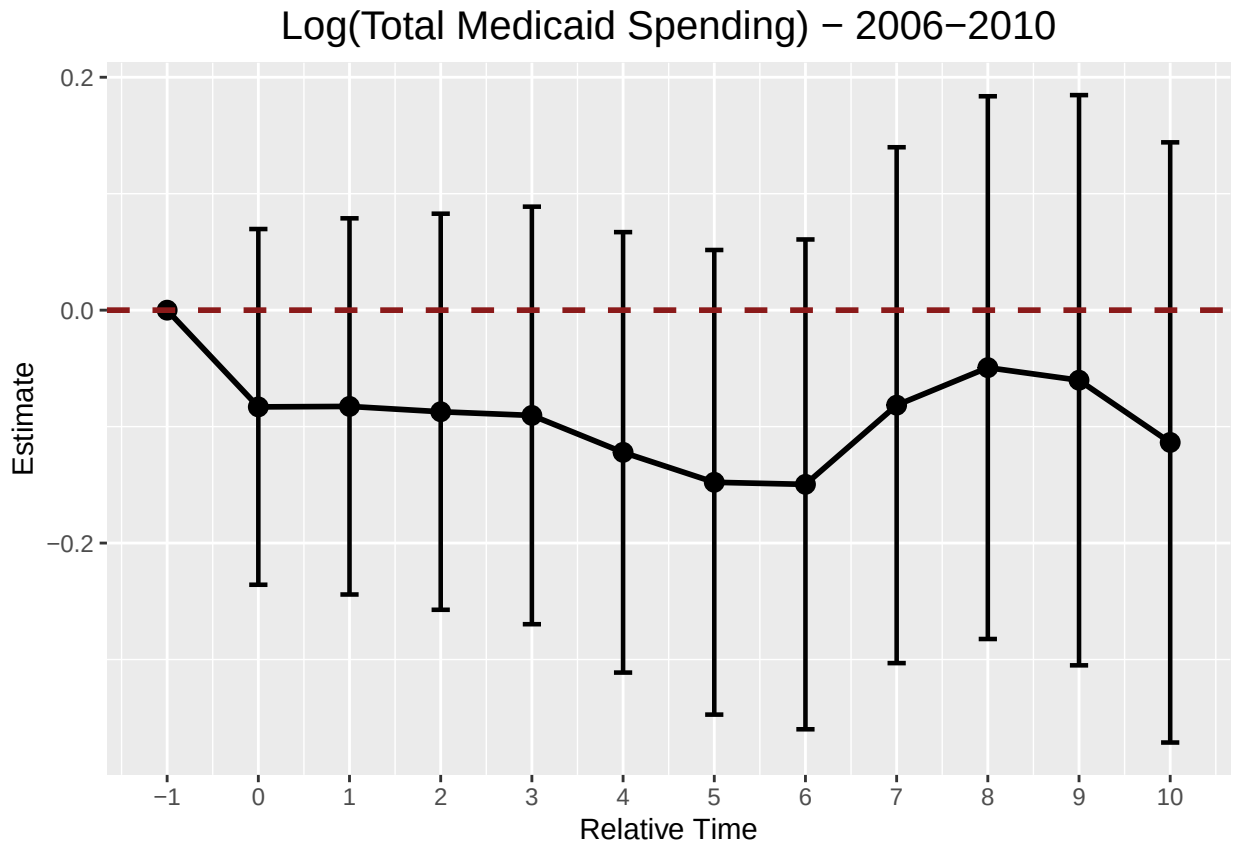

```
##
## $`log_total_spend_1996-2000`[[2]]
```



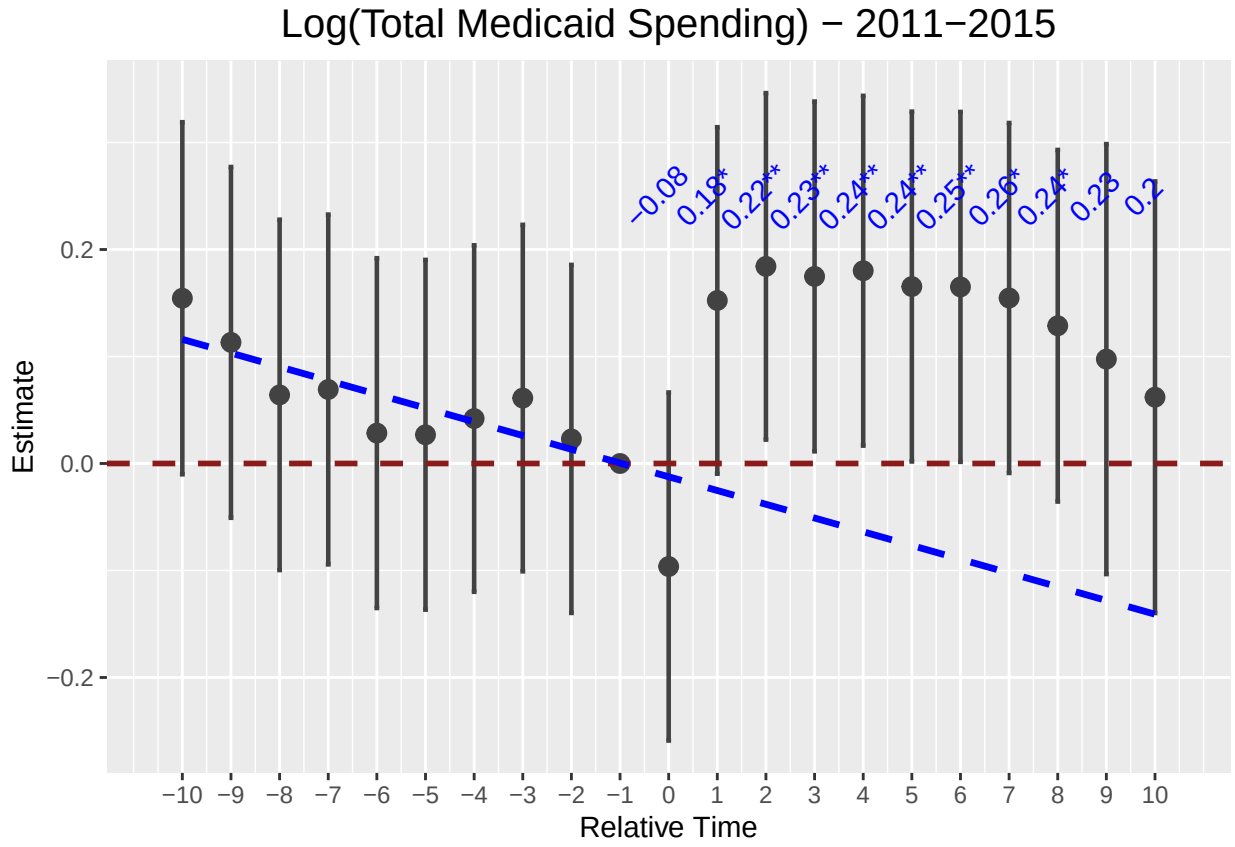

```
##
##
## $`log_total_spend_2006-2010`
## $`log_total_spend_2006-2010`[[1]]
```



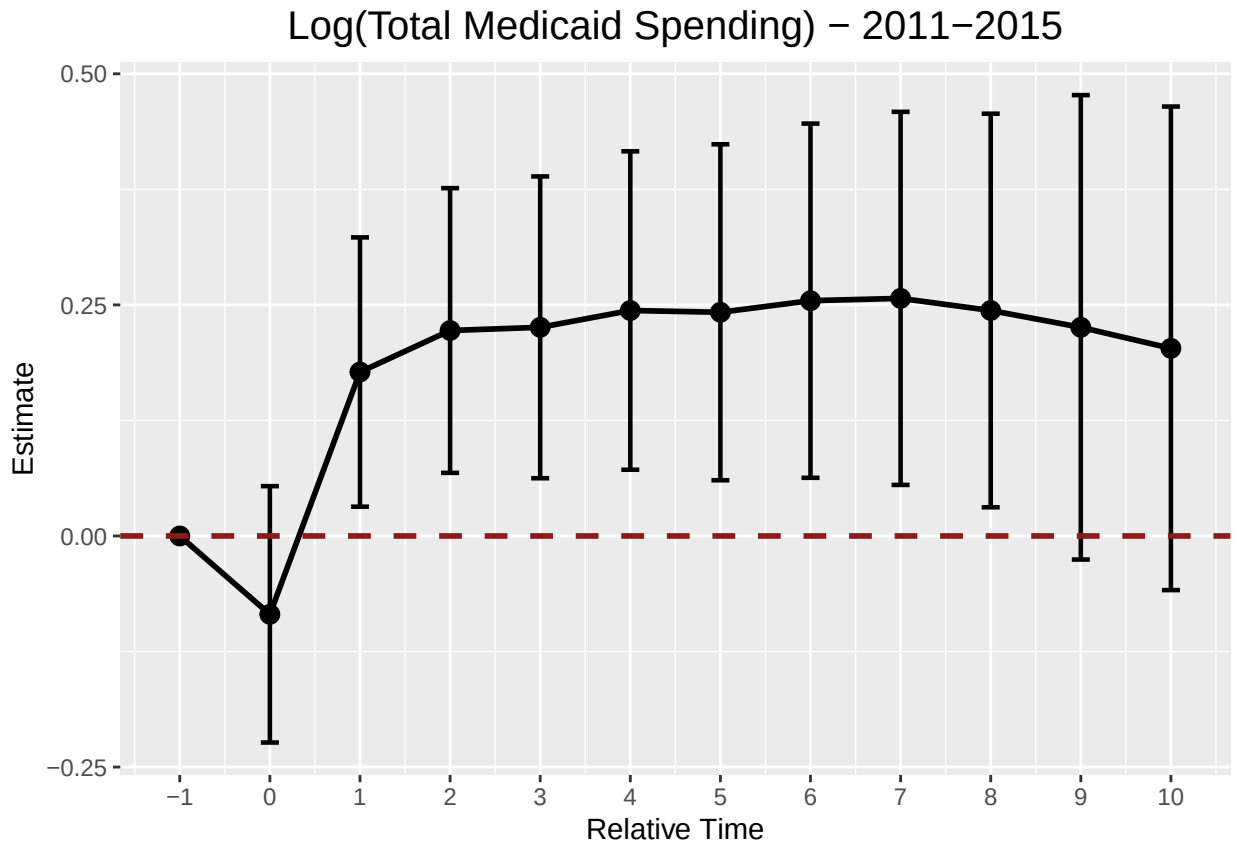

```
##
## $`log_total_spend_2006-2010`[[2]]
```



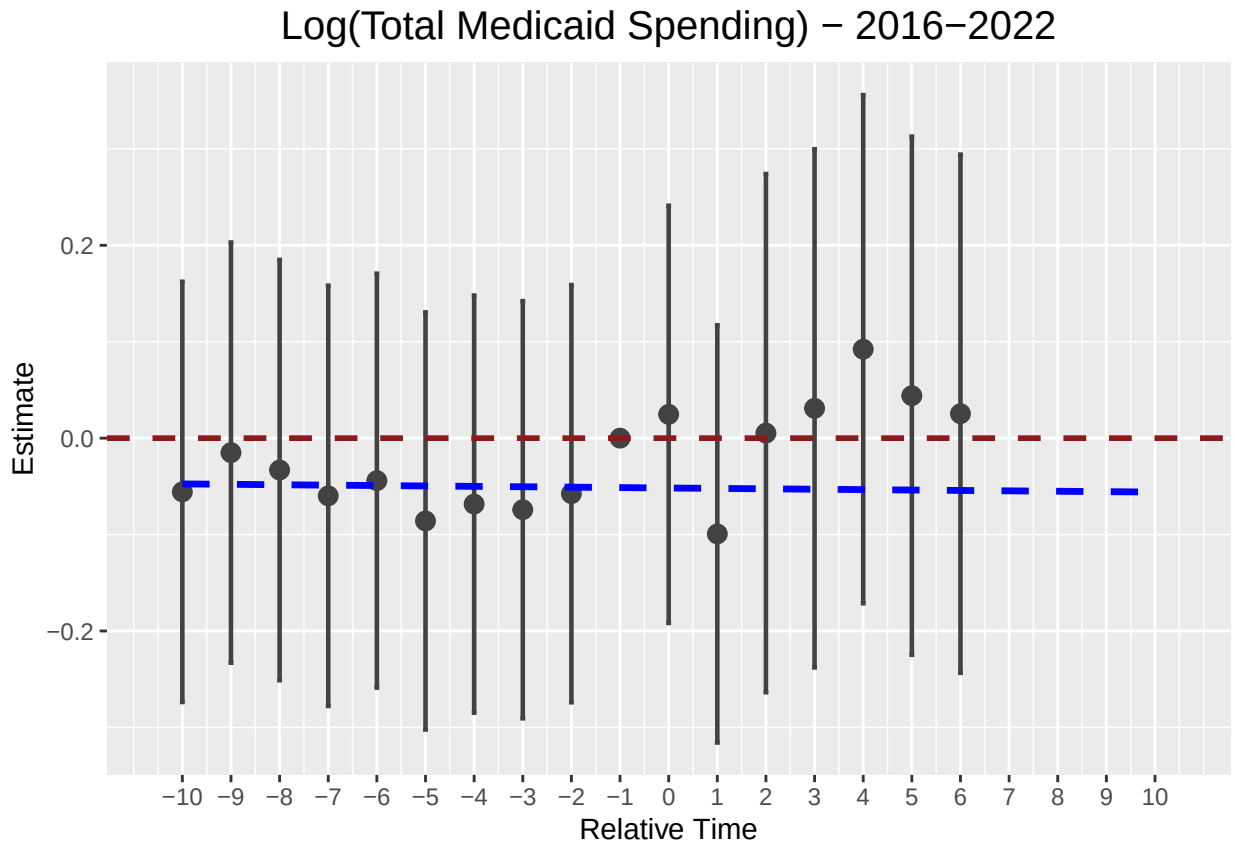

```
##
##
## $`log_total_spend_2011-2015`
## $`log_total_spend_2011-2015`[[1]]
```



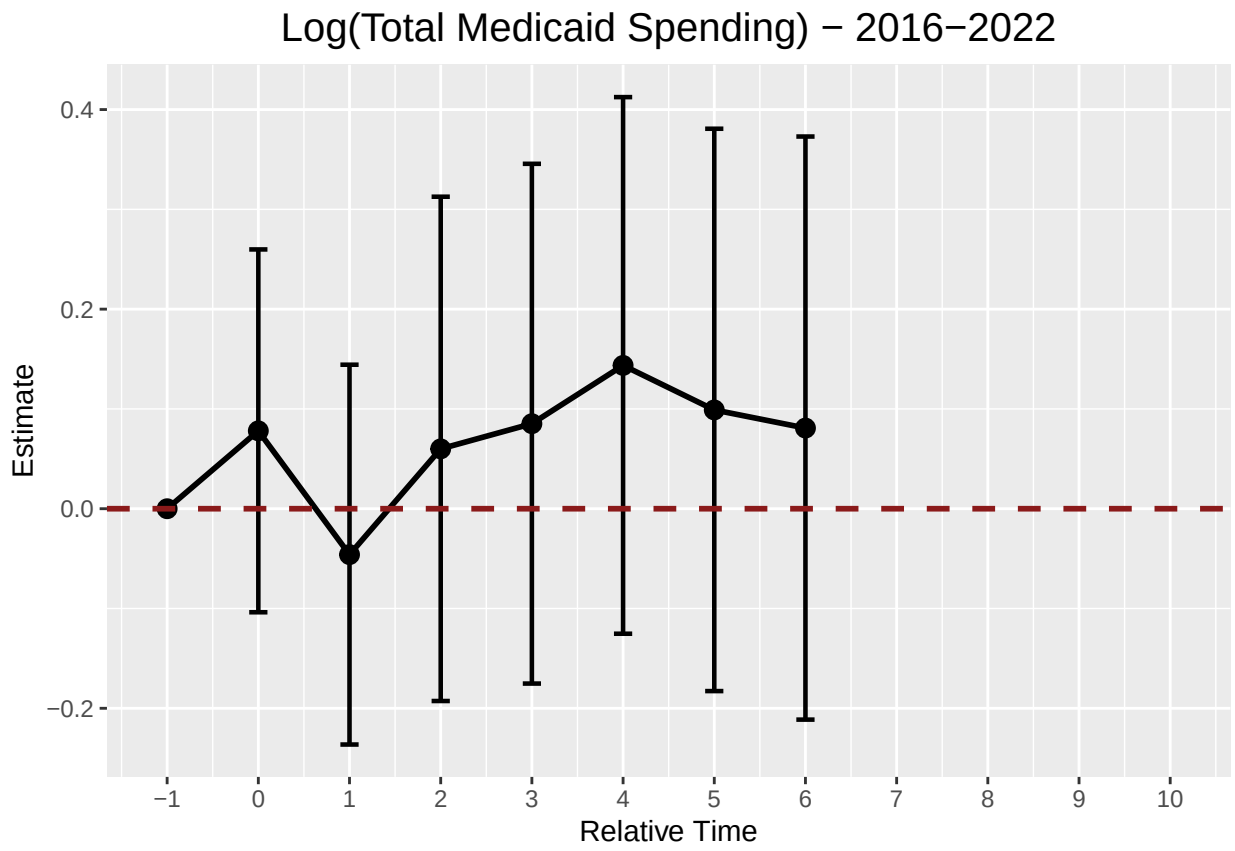

```
##
## $`log_total_spend_2011-2015`[[2]]
```



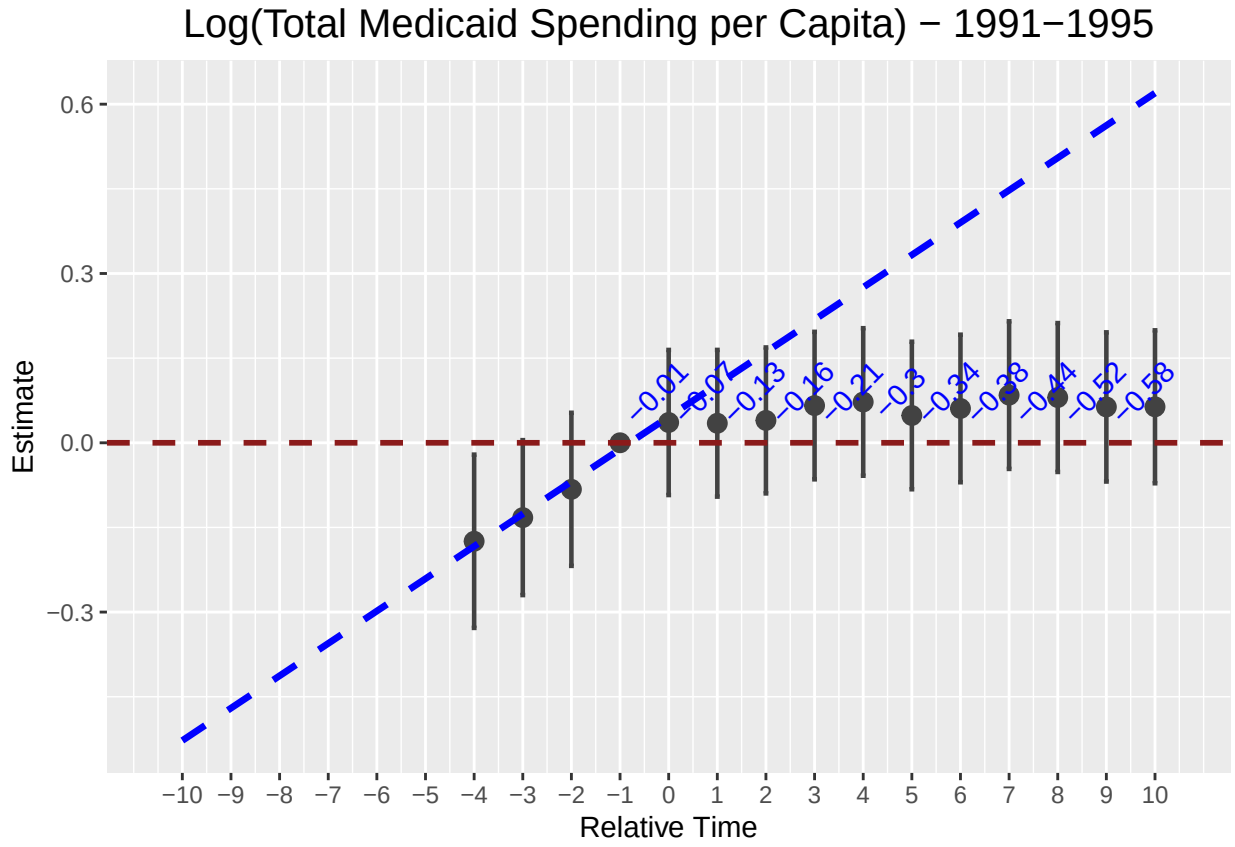

```
##
##
## $`log_total_spend_2016-2022`
## $`log_total_spend_2016-2022`[[1]]
```



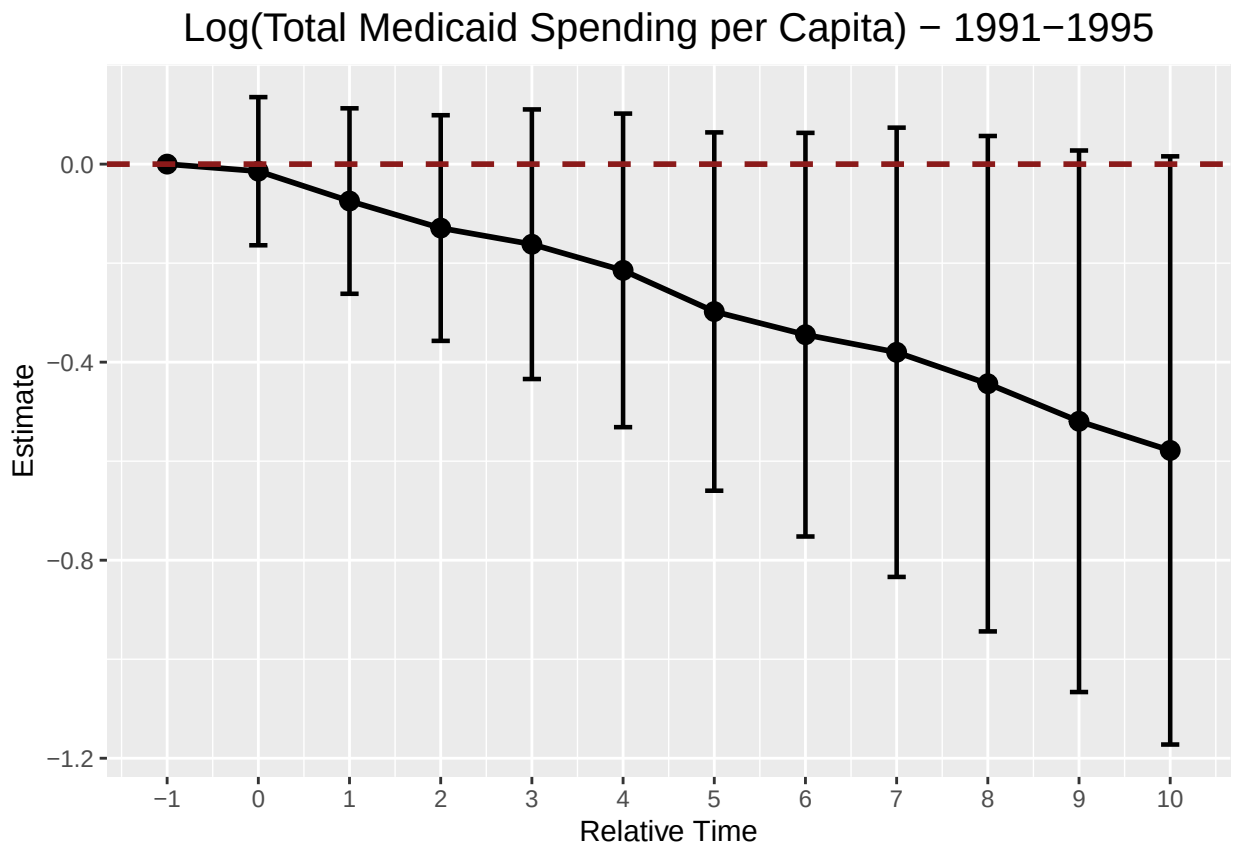

```
##
## $`log_total_spend_2016-2022`[[2]]
```



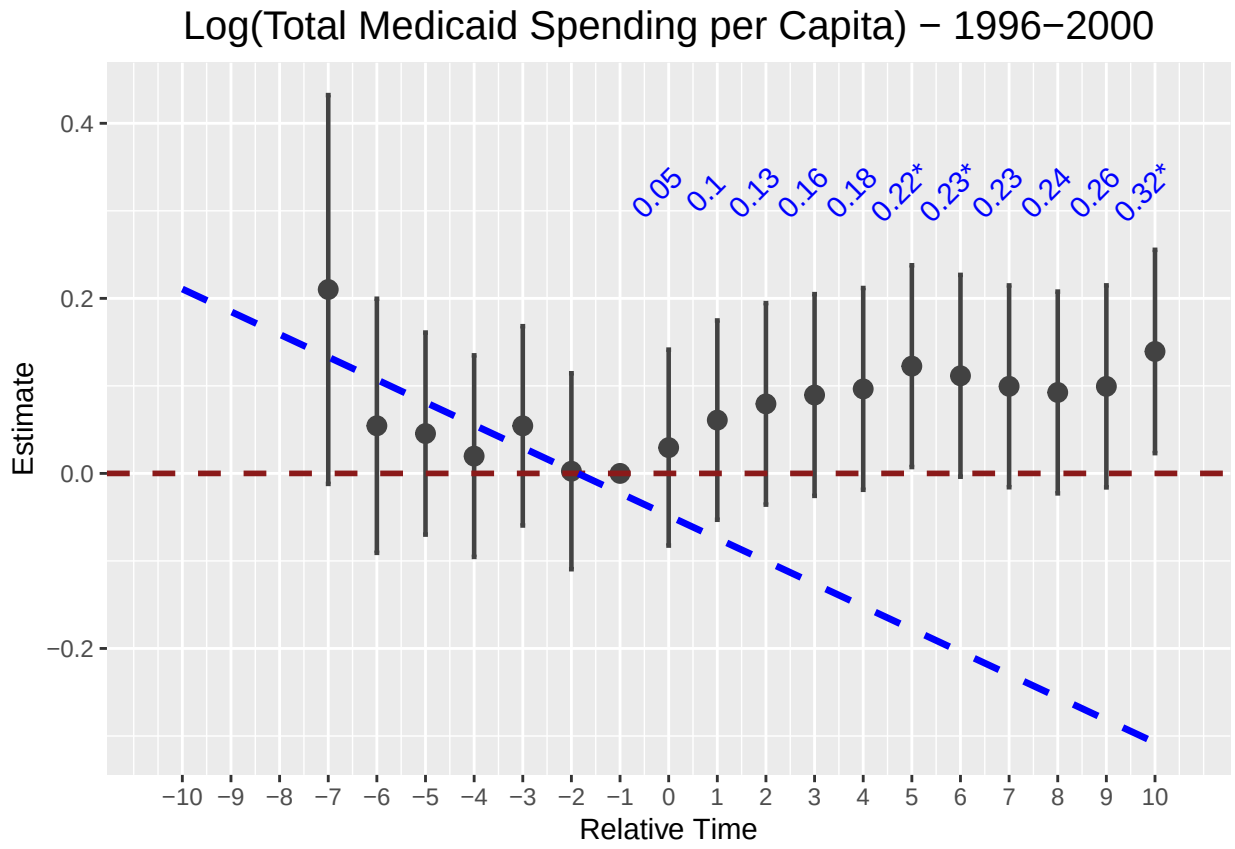

```
##
##
## $`log_all_medicaid_spending_per_cap_1991-1995`
## $`log_all_medicaid_spending_per_cap_1991-1995`[[1]]
```



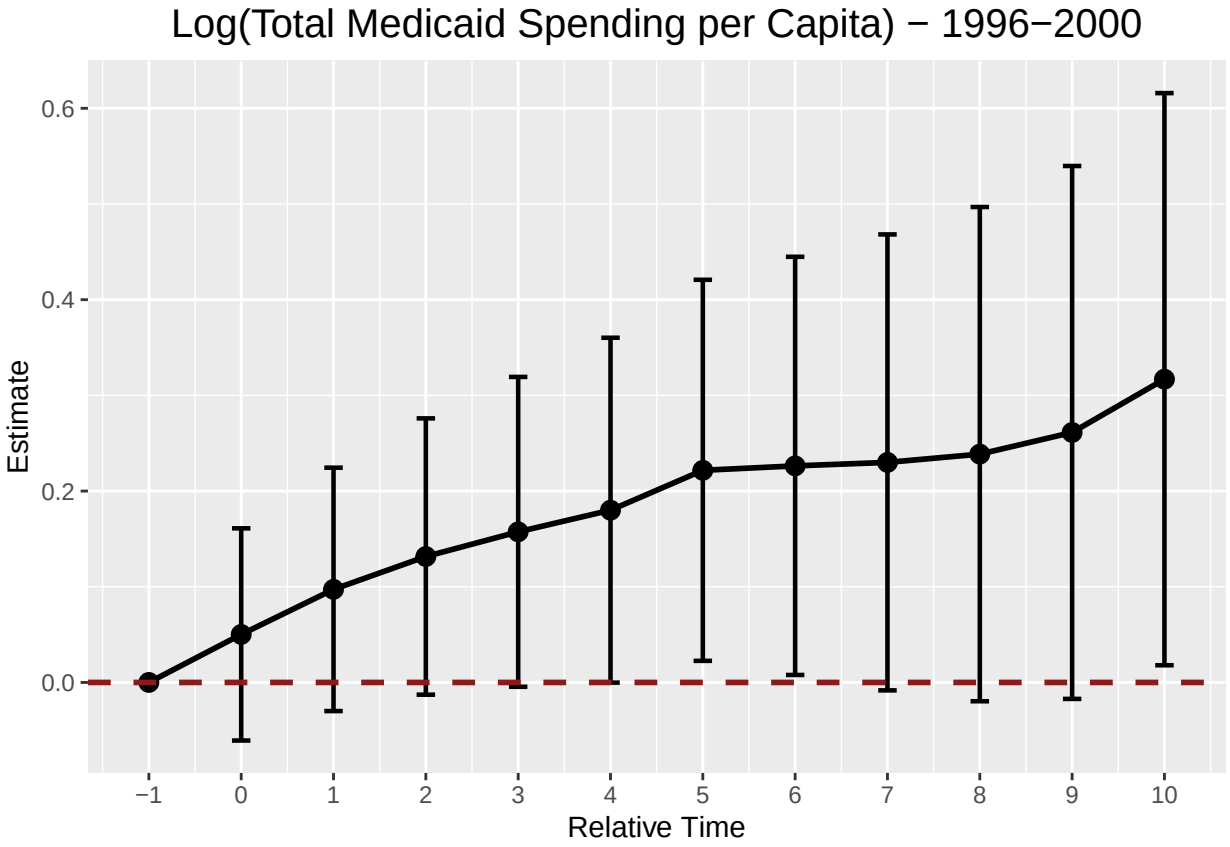

```
##
## $`log_all_medicaid_spending_per_cap_1991-1995`[[2]]
```



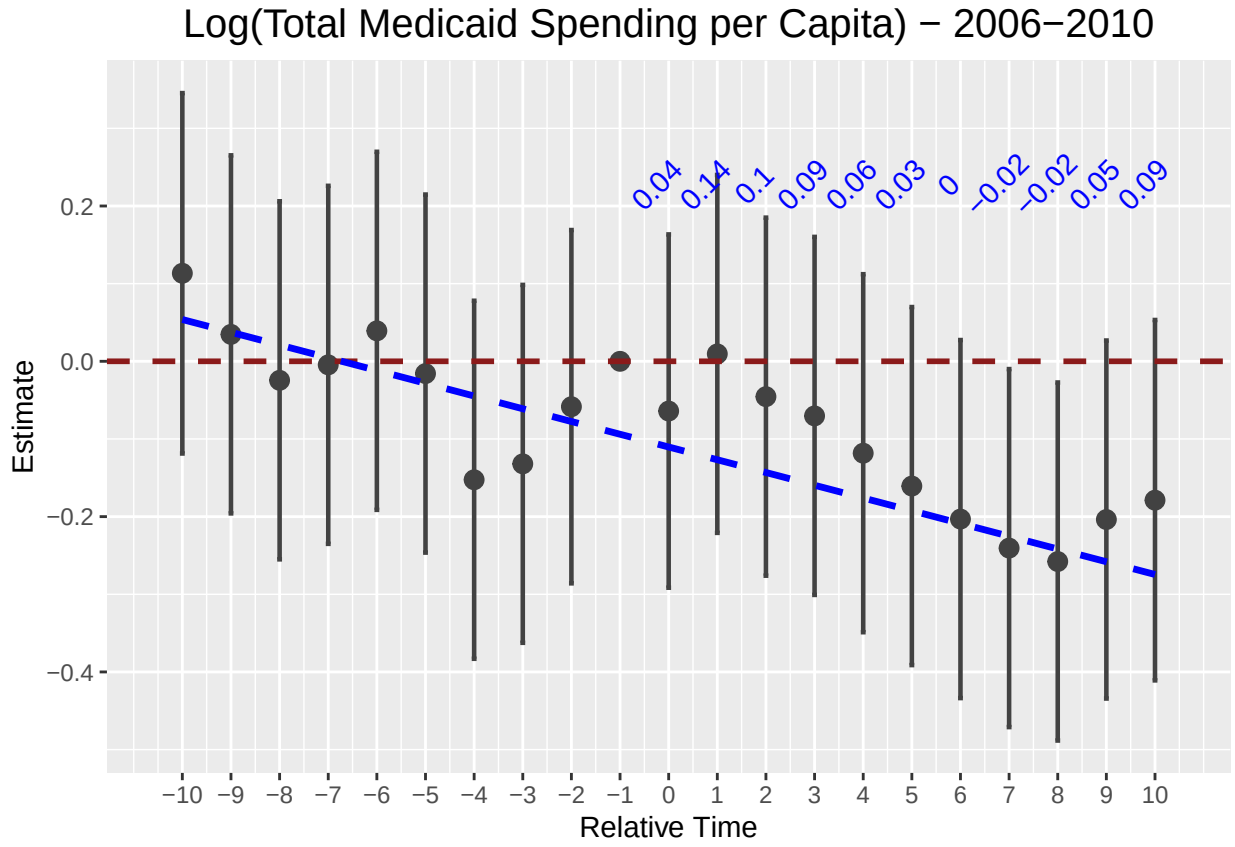

```
##
##
## $`log_all_medicaid_spending_per_cap_1996-2000`
## $`log_all_medicaid_spending_per_cap_1996-2000`[[1]]
```



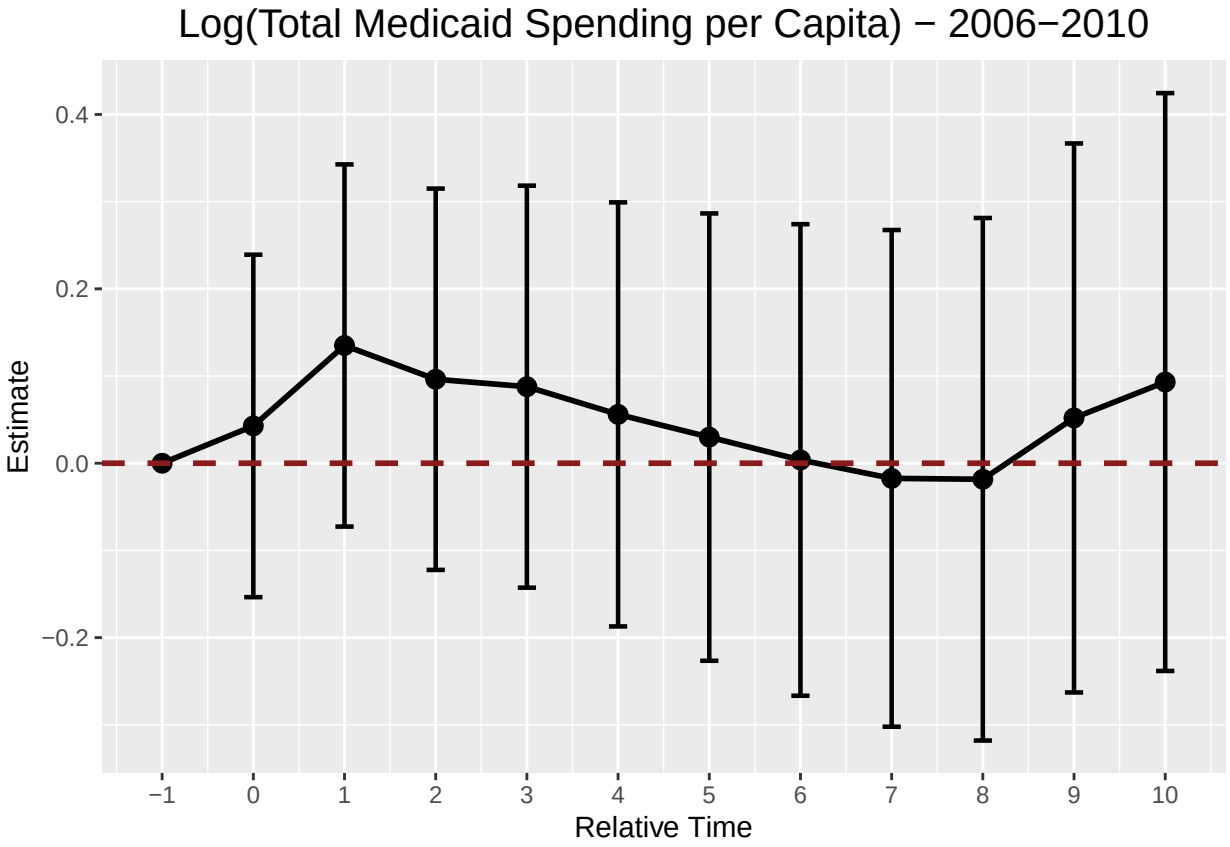

```
##
## $`log_all_medicaid_spending_per_cap_1996-2000`[[2]]
```



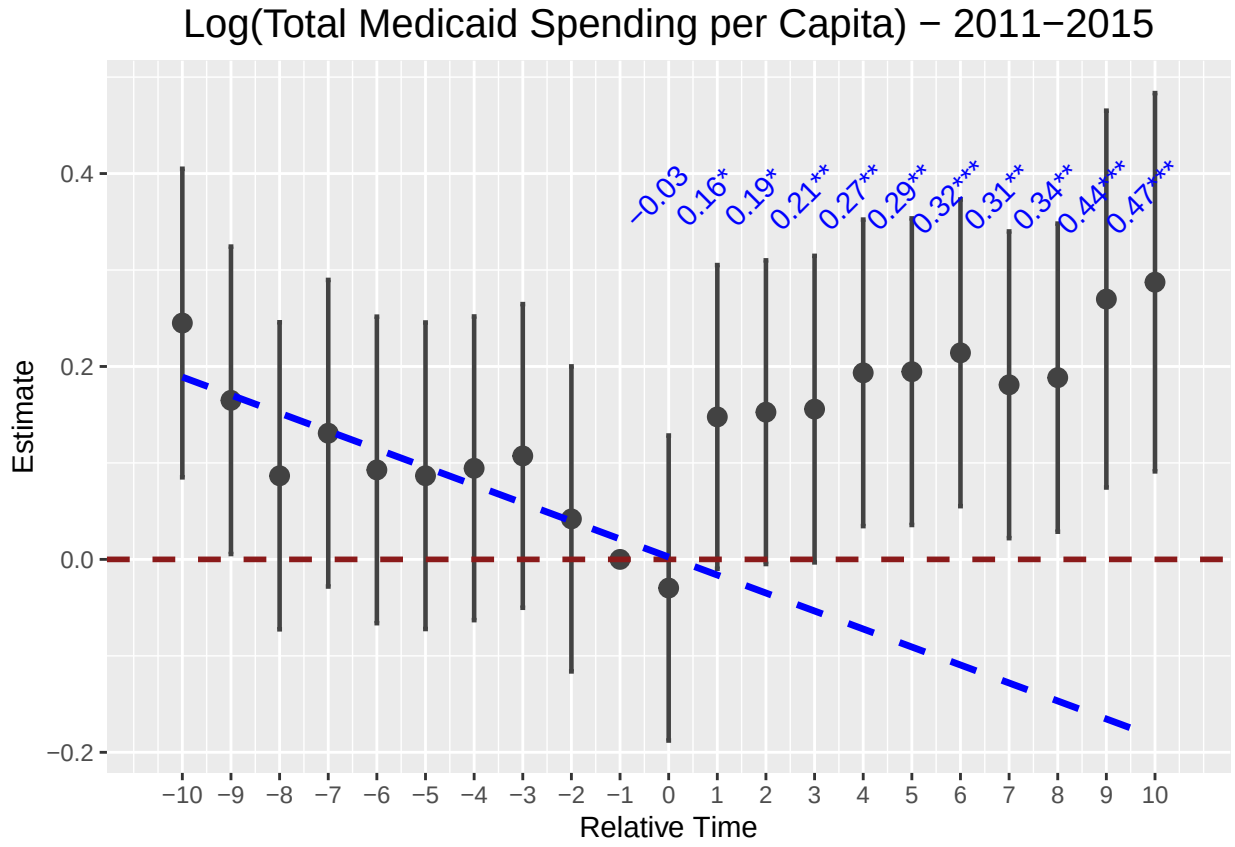

```
##
##
## $`log_all_medicaid_spending_per_cap_2006-2010`
## $`log_all_medicaid_spending_per_cap_2006-2010`[[1]]
```



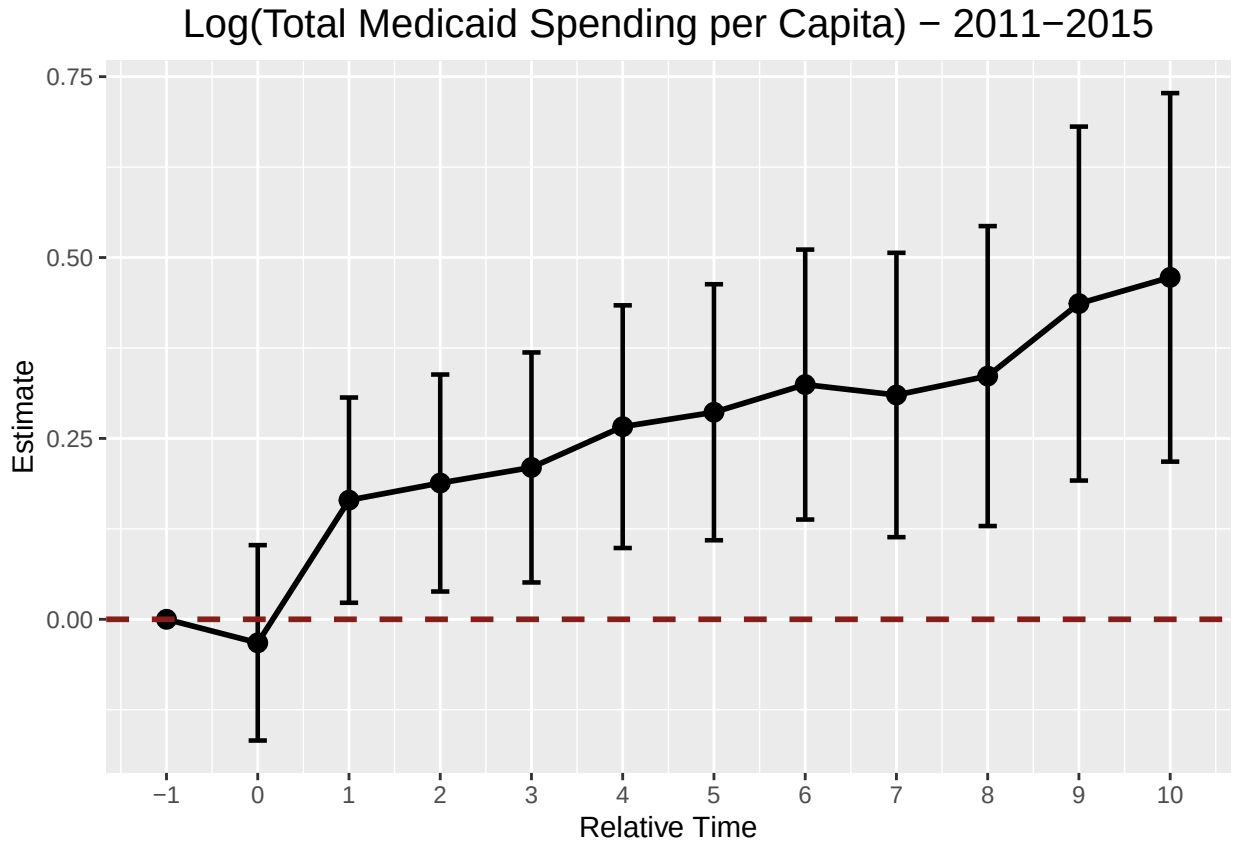

```
##
## $`log_all_medicaid_spending_per_cap_2006-2010`[[2]]
```



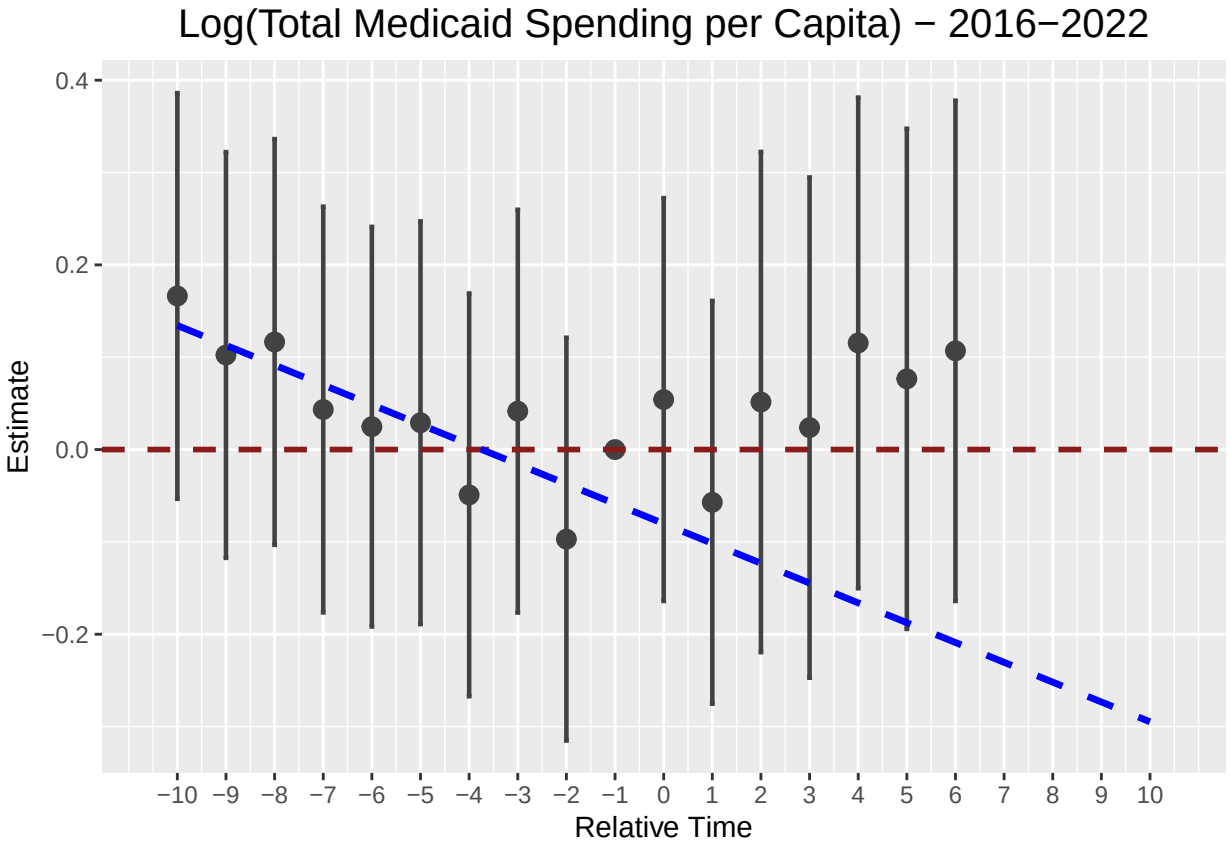

```
##
##
## $`log_all_medicaid_spending_per_cap_2011-2015`
## $`log_all_medicaid_spending_per_cap_2011-2015`[[1]]
```



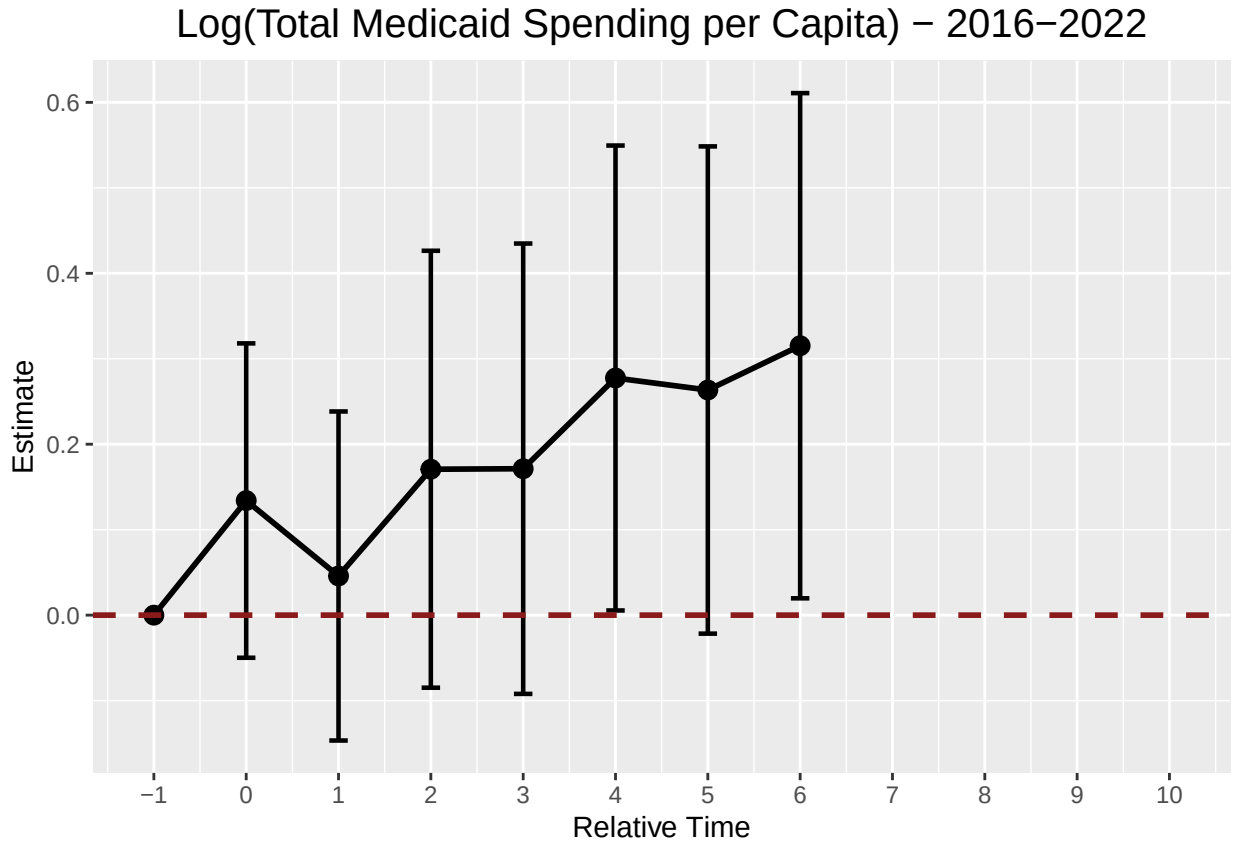

```
##
## $`log_all_medicaid_spending_per_cap_2011-2015`[[2]]
```




```
##
##
## $`log_all_medicaid_spending_per_cap_2016-2022`
## $`log_all_medicaid_spending_per_cap_2016-2022`[[1]]
```




```
##
## $`log_all_medicaid_spending_per_cap_2016-2022`[[2]]
```




```r
# Define treatment cohorts
assign_cohort_10 <- function(treatment_year) {
  case_when(
    treatment_year >= 1991 & treatment_year <= 2000 ~ "1991-2000",
    treatment_year >= 2001 & treatment_year <= 2010 ~ "2001-2010",
    treatment_year >= 2011 & treatment_year <= 2022 ~ "2011-2022",
    TRUE ~ "never-treated"  # Default fallback if none of the above match
  )
}

# Construct stacked data
# For each treatment year, construct control without treatment within 10/20 years
collect_10 <- list()
collect_20 <- list()

for(i in unique(data_cohorts$treatment_year[data_cohorts$treatment_year != Inf])){

  # Get treated state
  treated_state <- data_cohorts %>%
    filter(treatment_year == i) %>%
    mutate(treatment_year_group = i) %>%
    mutate(cohort_10_interval = assign_cohort_10(treatment_year_group)) %>%
    select(state, year, center_time, treated, total_medicaid,
           log_all_medicaid_spending_per_cap, all_medicaid_spending_per_cap,
           pct_in_comp_mco, treatment_year, treatment_year_group, cohort_10_interval)
```

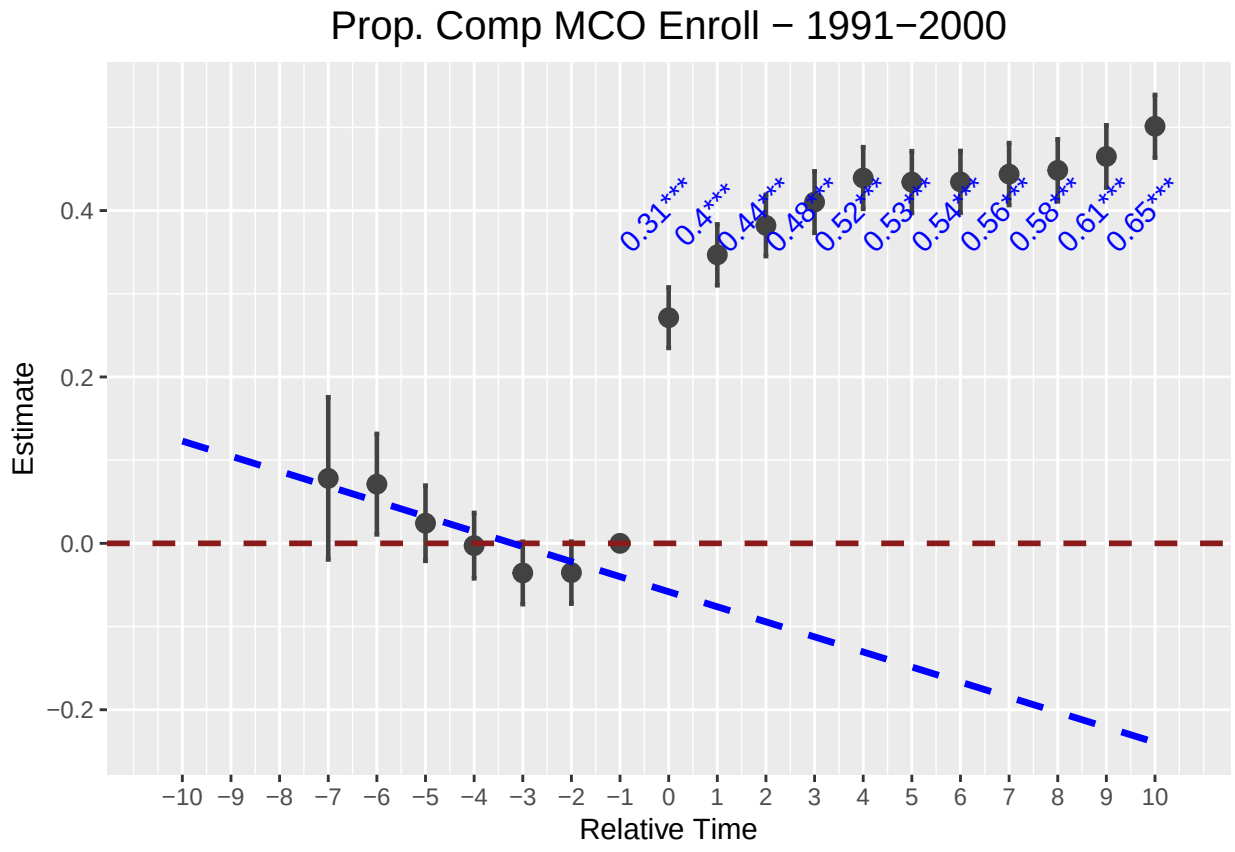
```
  max_year <- max(treated_state$year)

  # Get control states within a 10 year (pre and post) from treatment year
  control_state_10 <- data_cohorts %>%
    filter(treatment_year > i + 10,
           year <= i + 10, year >= i - 10) %>%
    mutate(treatment_year_group = i,
           center_time = 0,
           treated = 0,
           cohort_10_interval = assign_cohort_10(treatment_year_group)) %>%
    select(state, year, center_time, treated, total_medicaid,
           log_all_medicaid_spending_per_cap, all_medicaid_spending_per_cap,
           pct_in_comp_mco, treatment_year, treatment_year_group, cohort_10_interval)

  control_state_20 <- data_cohorts %>%
    filter(treatment_year > i + 20,
           year <= i + 20, year >= i - 20) %>%
    mutate(treatment_year_group = i,
           center_time = 0,
           treated = 0,
           cohort_10_interval = assign_cohort_10(treatment_year_group)) %>%
    select(state, year, center_time, treated, total_medicaid,
           log_all_medicaid_spending_per_cap, all_medicaid_spending_per_cap,
           pct_in_comp_mco, treatment_year, treatment_year_group, cohort_10_interval)

  collect_10[[length(collect_10)+1]] <- rbind(treated_state, control_state_10)
  collect_20[[length(collect_20)+1]] <- rbind(treated_state, control_state_20)
}

dat_10 <- do.call(rbind, collect_10)
dat_20 <- do.call(rbind, collect_20)

dat_10 <- dat_10 %>% dummy_cols(select_columns = "center_time")
dat_20 <- dat_20 %>% dummy_cols(select_columns = "center_time")

# DID - 10 period
dat <- dat_10
dat <- dat %>%
  mutate(log_total_spend = log(total_medicaid))
```

Plots for 10 year interval cohorts

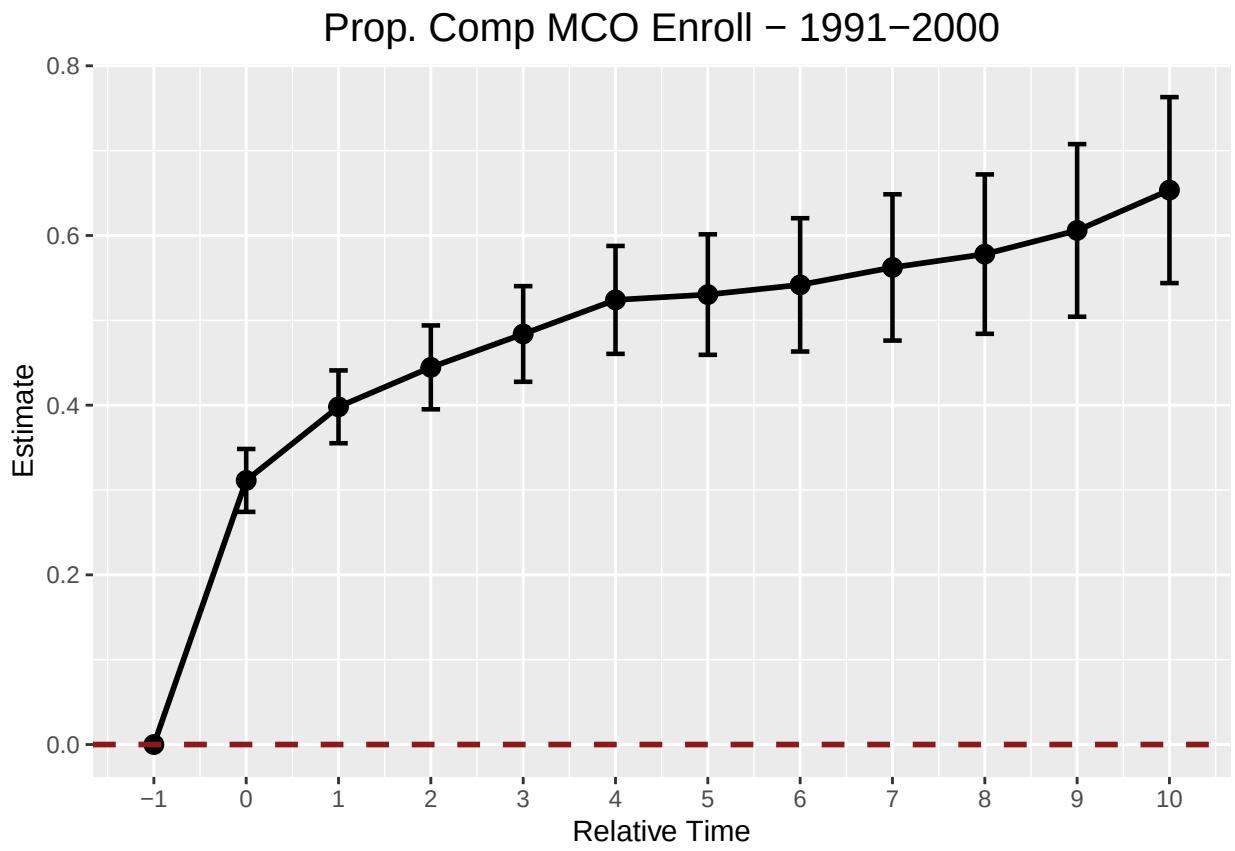
```
cohorts2 <- c("1991-2000", "2001-2010", "2011-2022")

outcomes <- list(
  pct_in_comp_mco = "Prop. Comp MCO Enroll",
  log_total_spend = "Log(Total Medicaid Spending)",
  log_all_medicaid_spending_per_cap = "Log(Total Medicaid Spending per Capita)"
)
```

```r
# Initialize an empty list to store the plots
plot_list <- list()

# Loop through each outcome
for (outcome_name in names(outcomes)) {
  outcome_label <- outcomes[[outcome_name]]

  # Loop through each cohort
  for (cohort in cohorts2) {
    # Generate the plot
    p <- did_graph_trend_10(dat, outcome_name, outcome_label, "cohort_10_interval", cohort)

    # Store the plot in the list
    plot_list[[paste(outcome_name, cohort, sep = "_")]] <- p
  }

  outcome_plots <- plot_list[grep(outcome_name, names(plot_list))]
  print(outcome_plots)
}
```

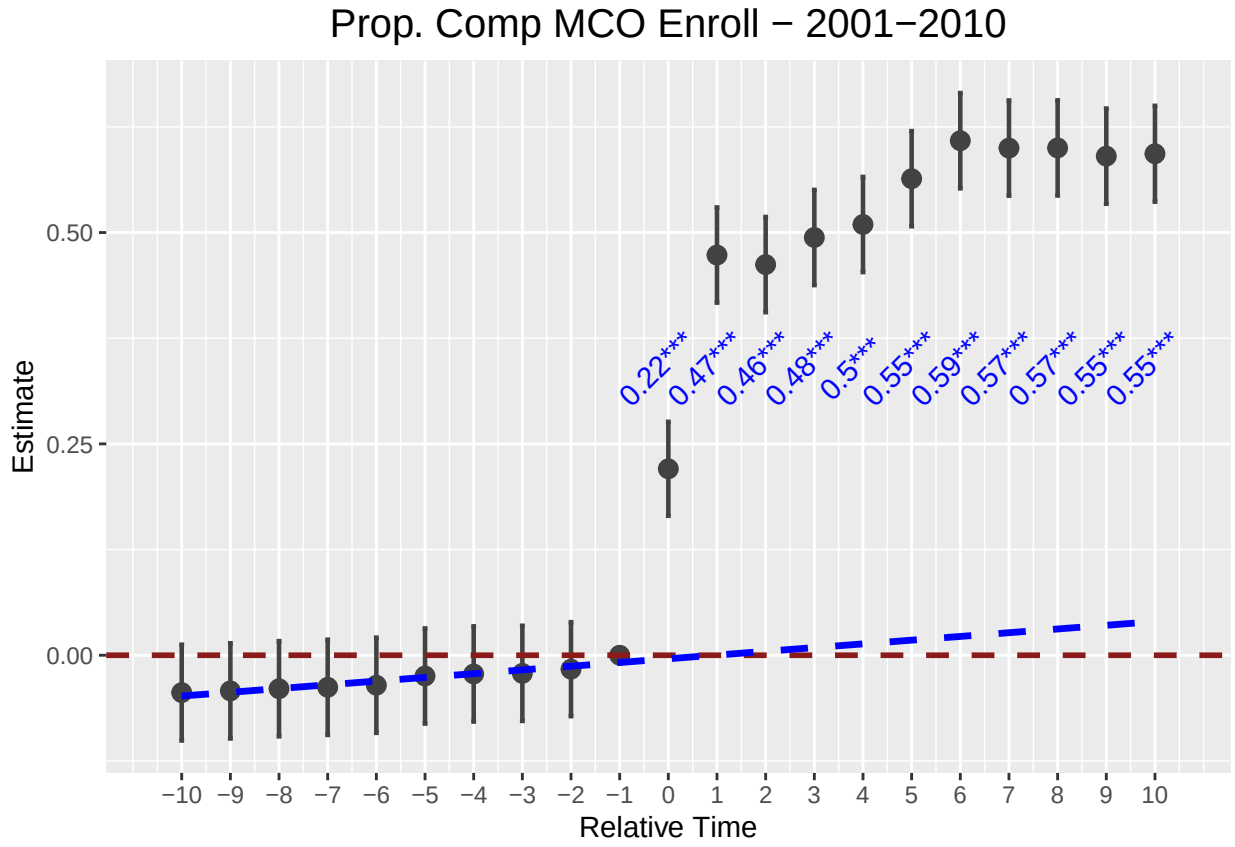
```
## $`pct_in_comp_mco_1991-2000`
## $`pct_in_comp_mco_1991-2000`[[1]]
```



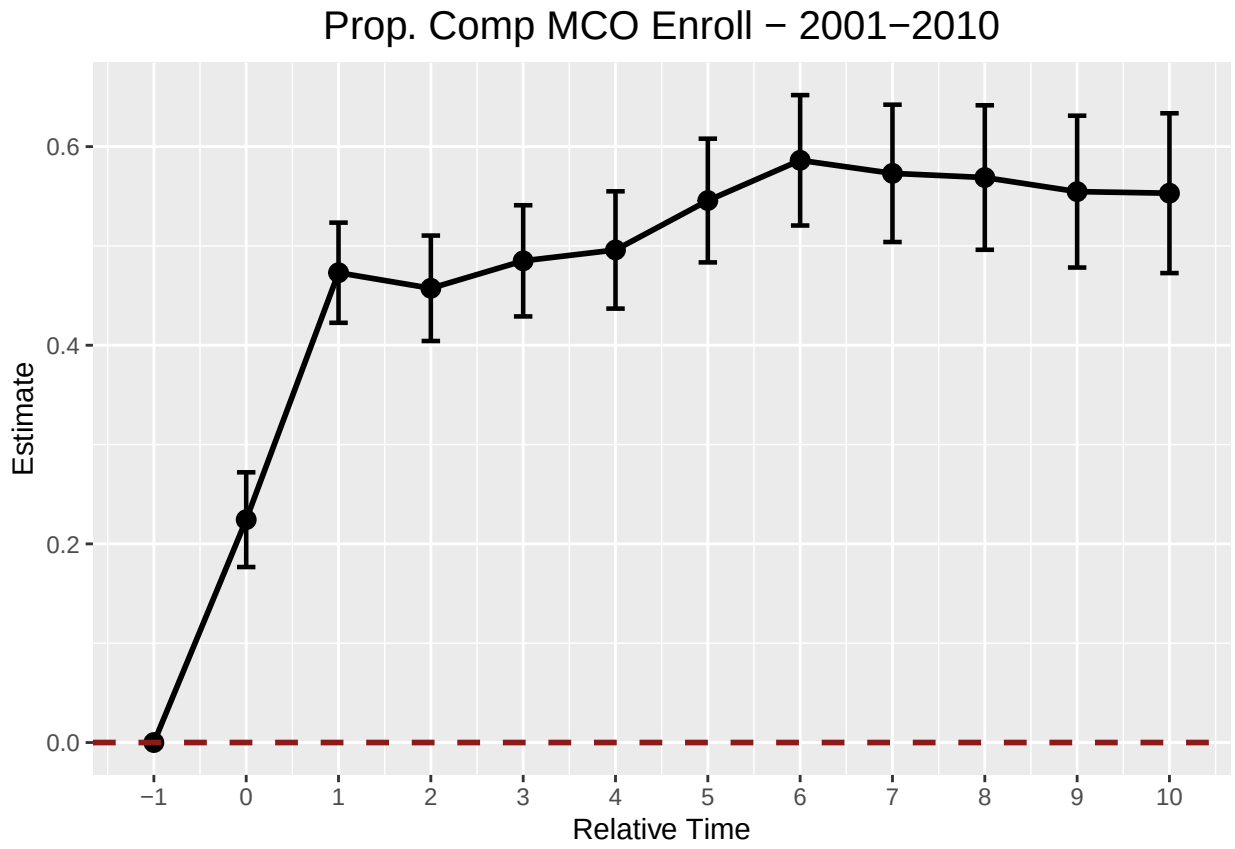

```
##
## $`pct_in_comp_mco_1991-2000`[[2]]
```



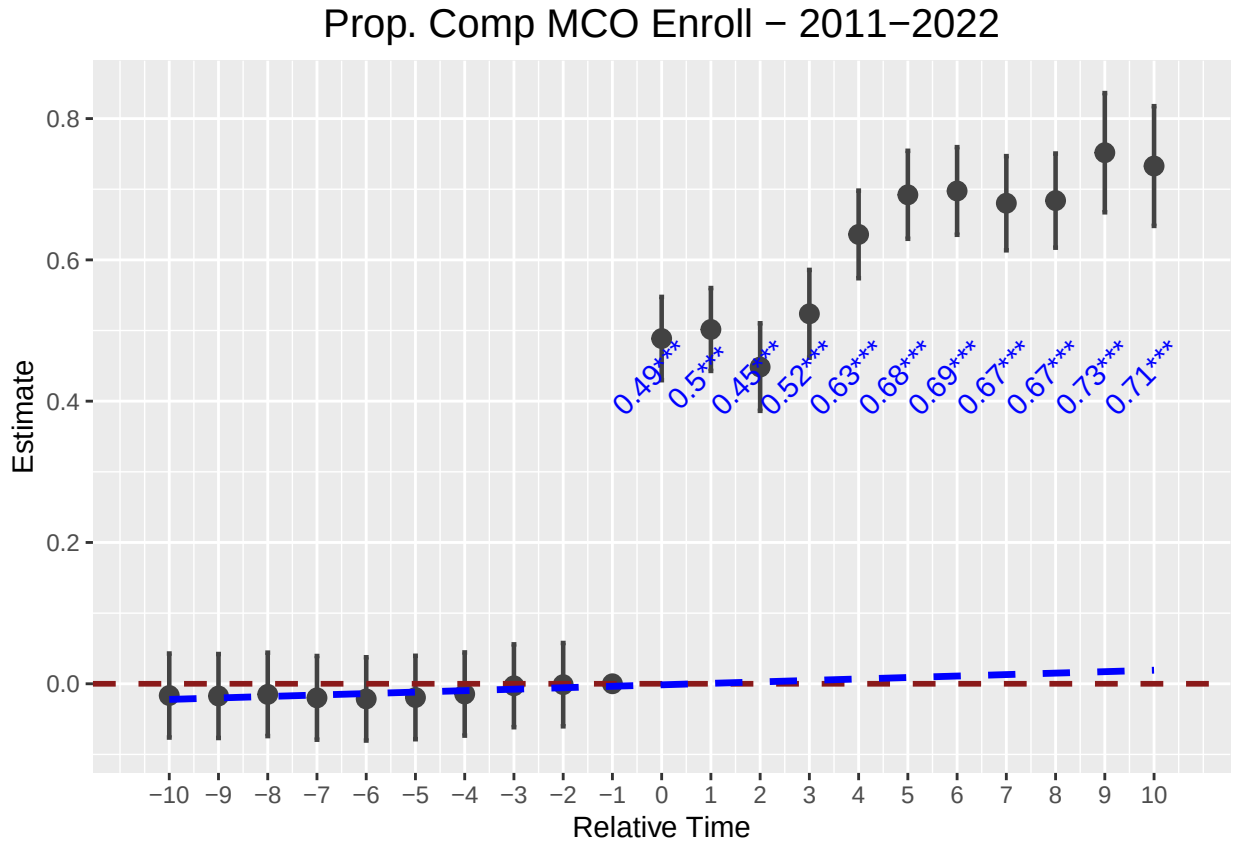

```
##
##
## $`pct_in_comp_mco_2001-2010`
## $`pct_in_comp_mco_2001-2010`[[1]]
```



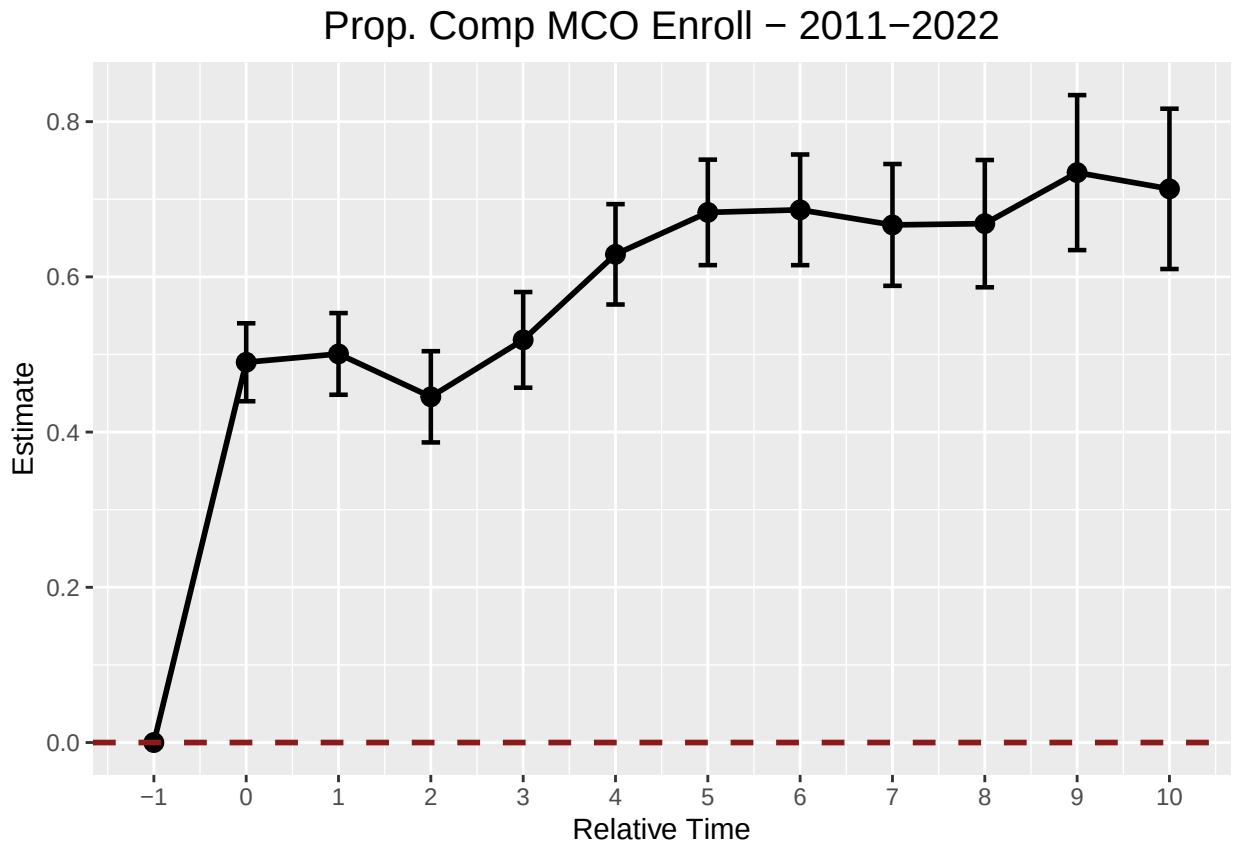

```
##
## $`pct_in_comp_mco_2001-2010`[[2]]
```



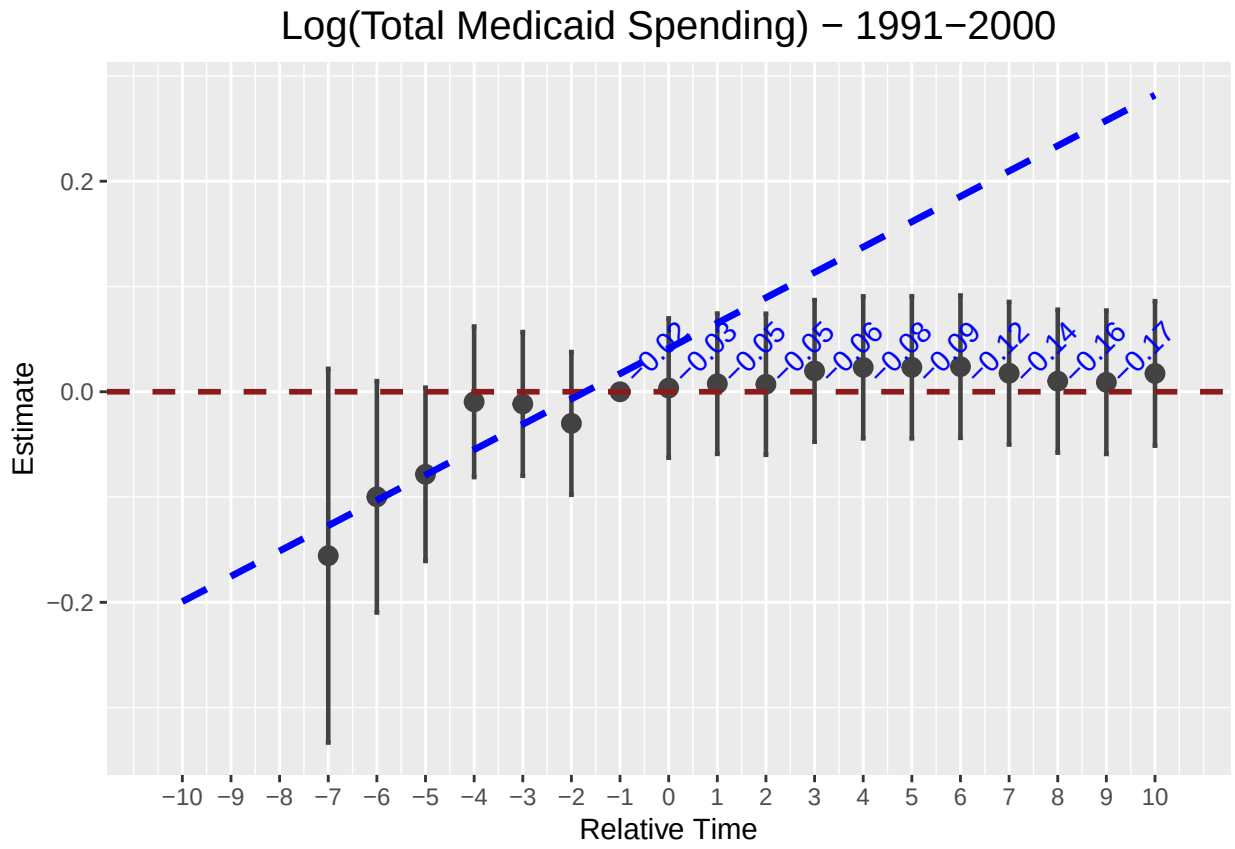

```
##
##
## $`pct_in_comp_mco_2011-2022`
## $`pct_in_comp_mco_2011-2022`[[1]]
```



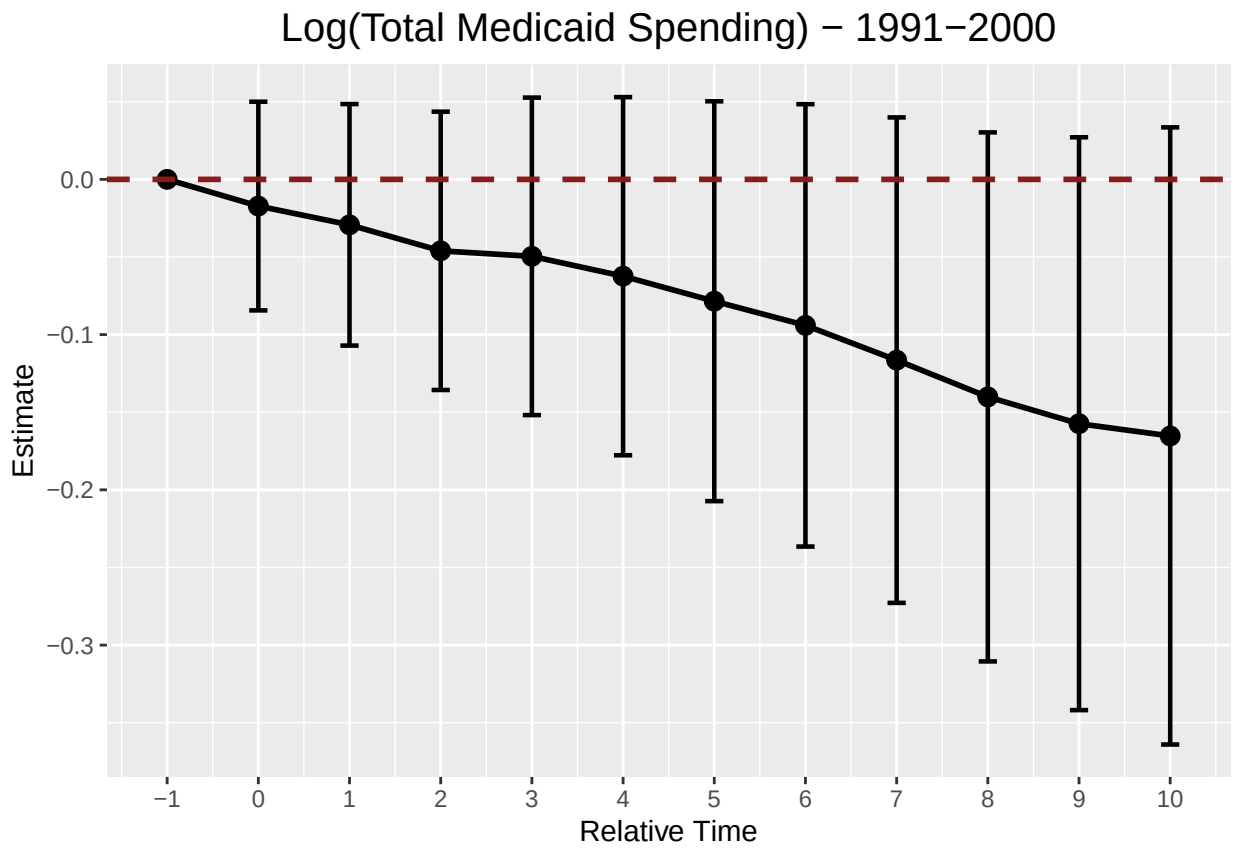

```
##
## $`pct_in_comp_mco_2011-2022`[[2]]
```



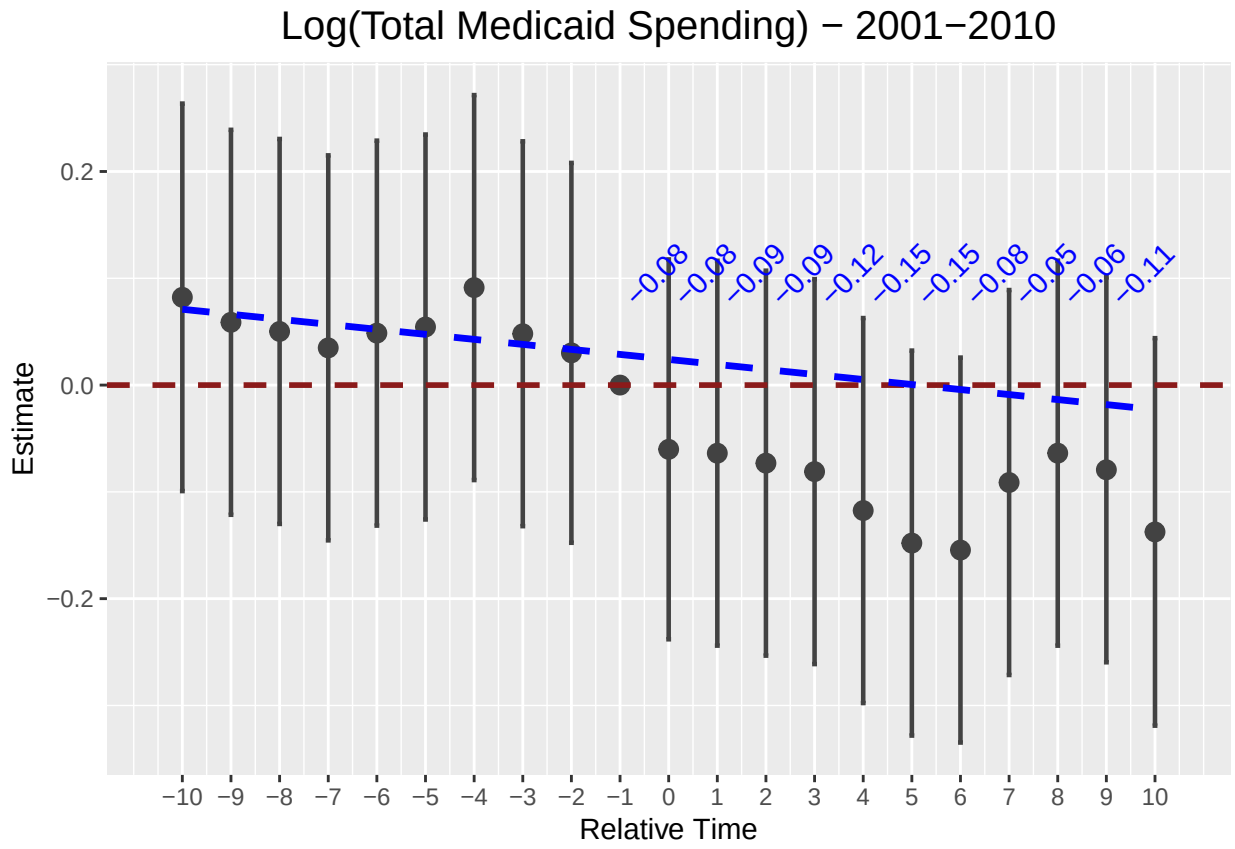

```
##
##
## $`log_total_spend_1991-2000`
## $`log_total_spend_1991-2000`[[1]]
```



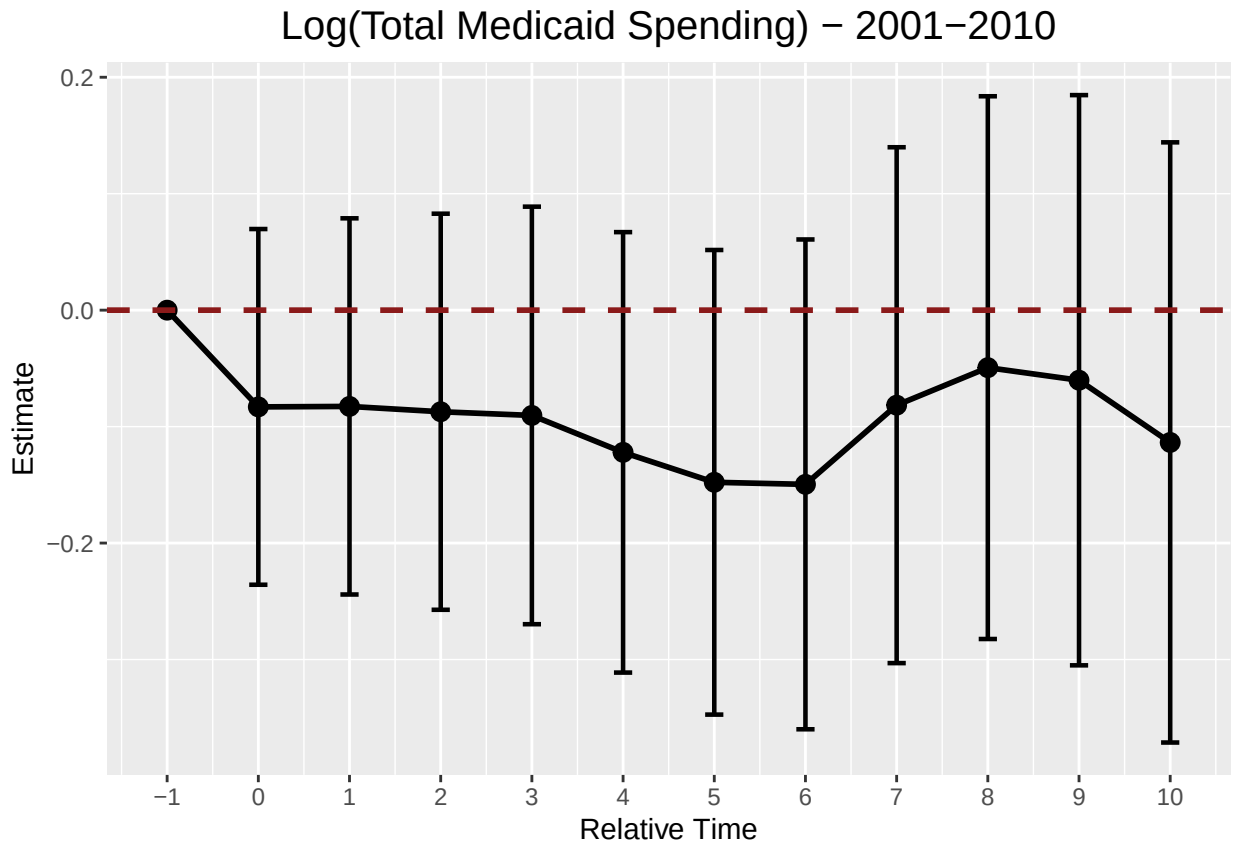

```
##
## $`log_total_spend_1991-2000`[[2]]
```



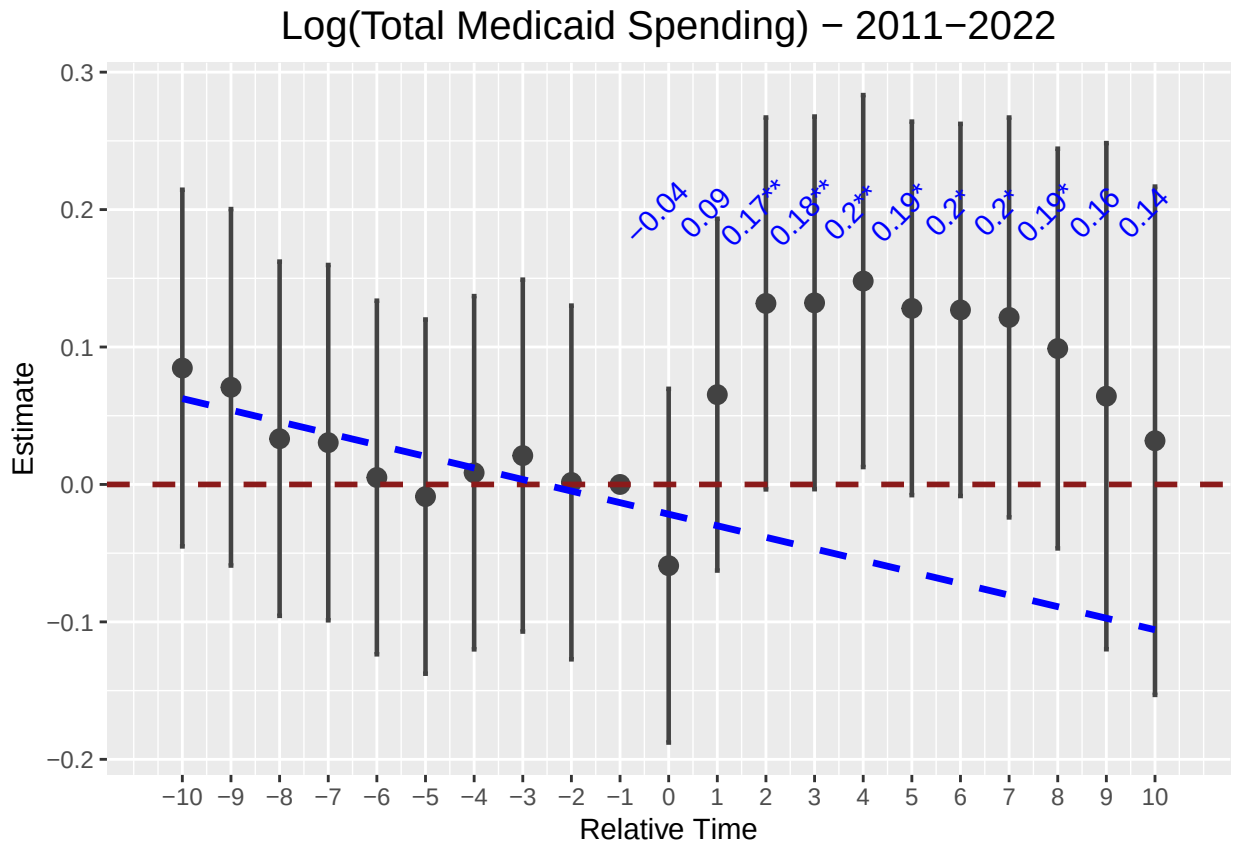

```
##
##
## $`log_total_spend_2001-2010`
## $`log_total_spend_2001-2010`[[1]]
```



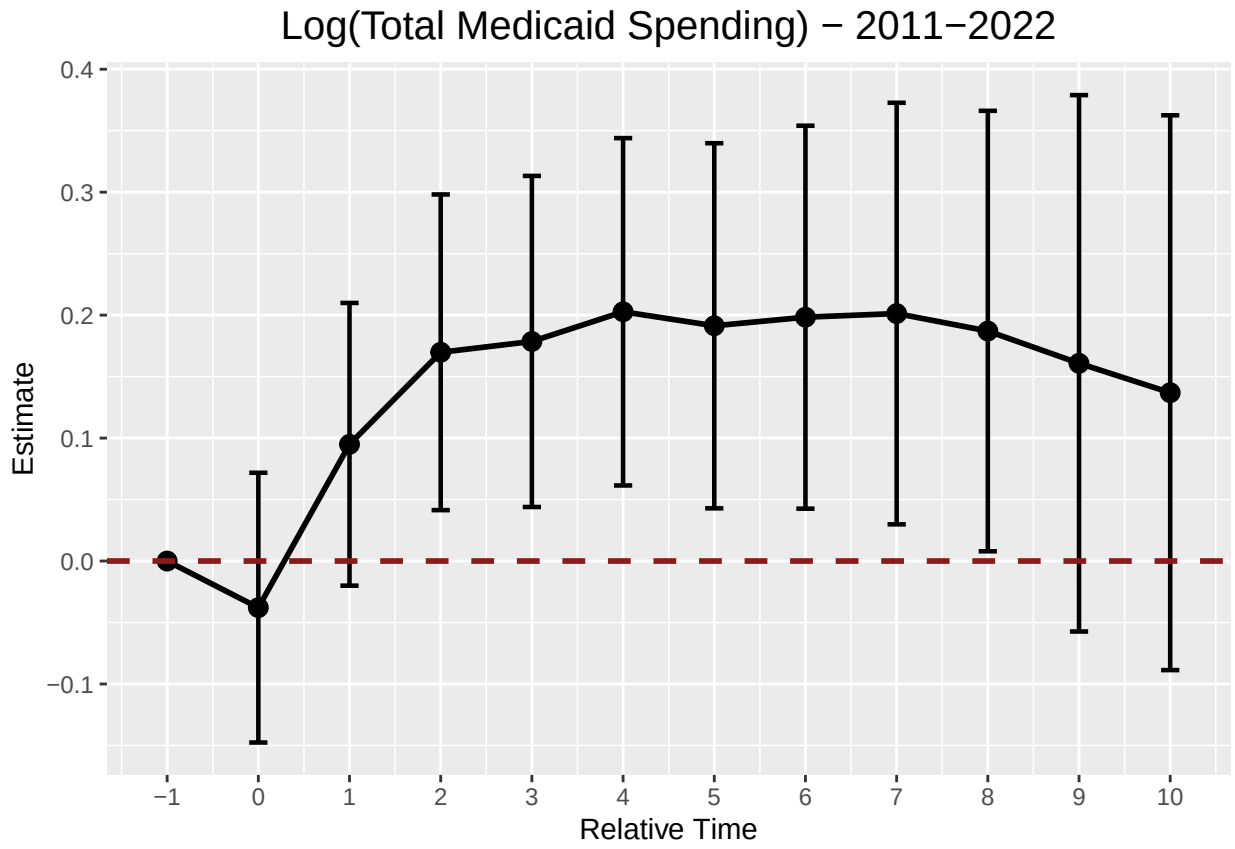

```
##
## $`log_total_spend_2001-2010`[[2]]
```



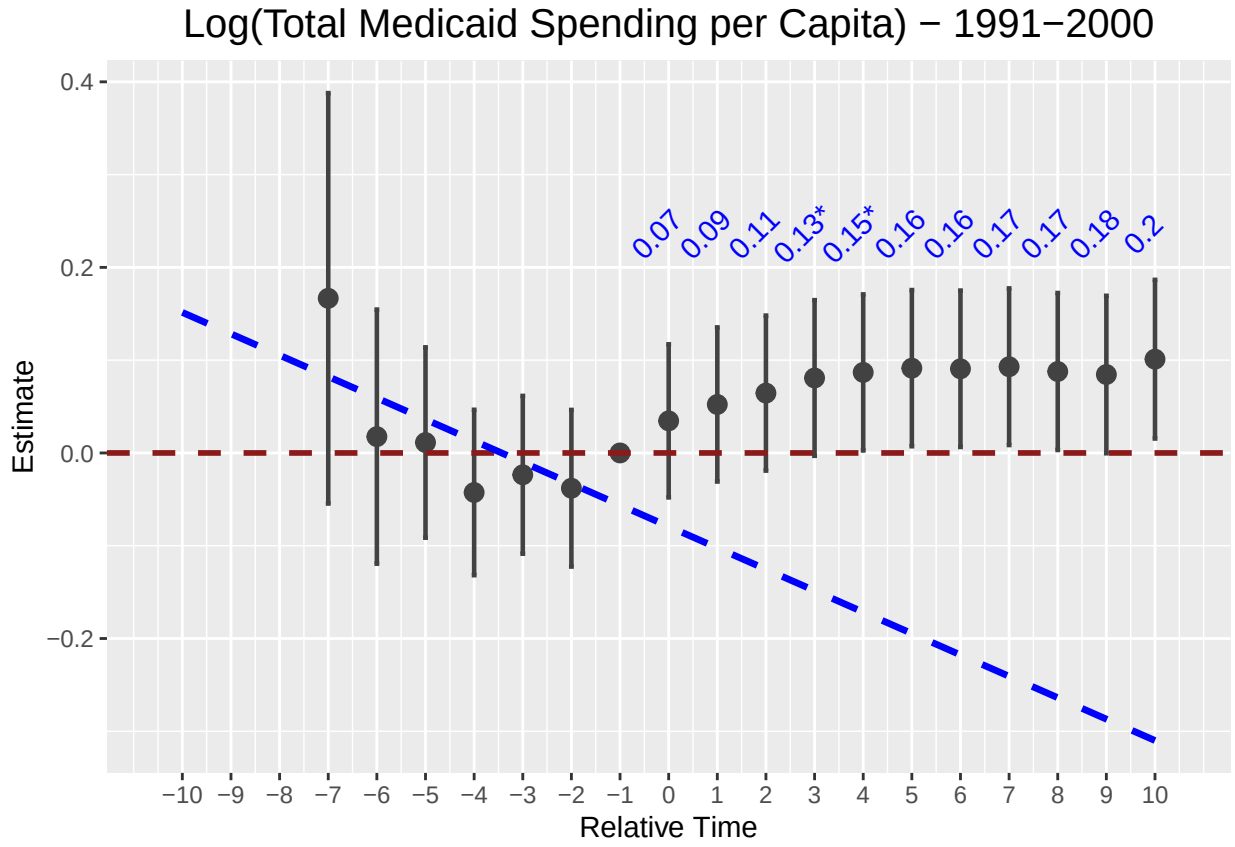

```
##
##
## $`log_total_spend_2011-2022`
## $`log_total_spend_2011-2022`[[1]]
```



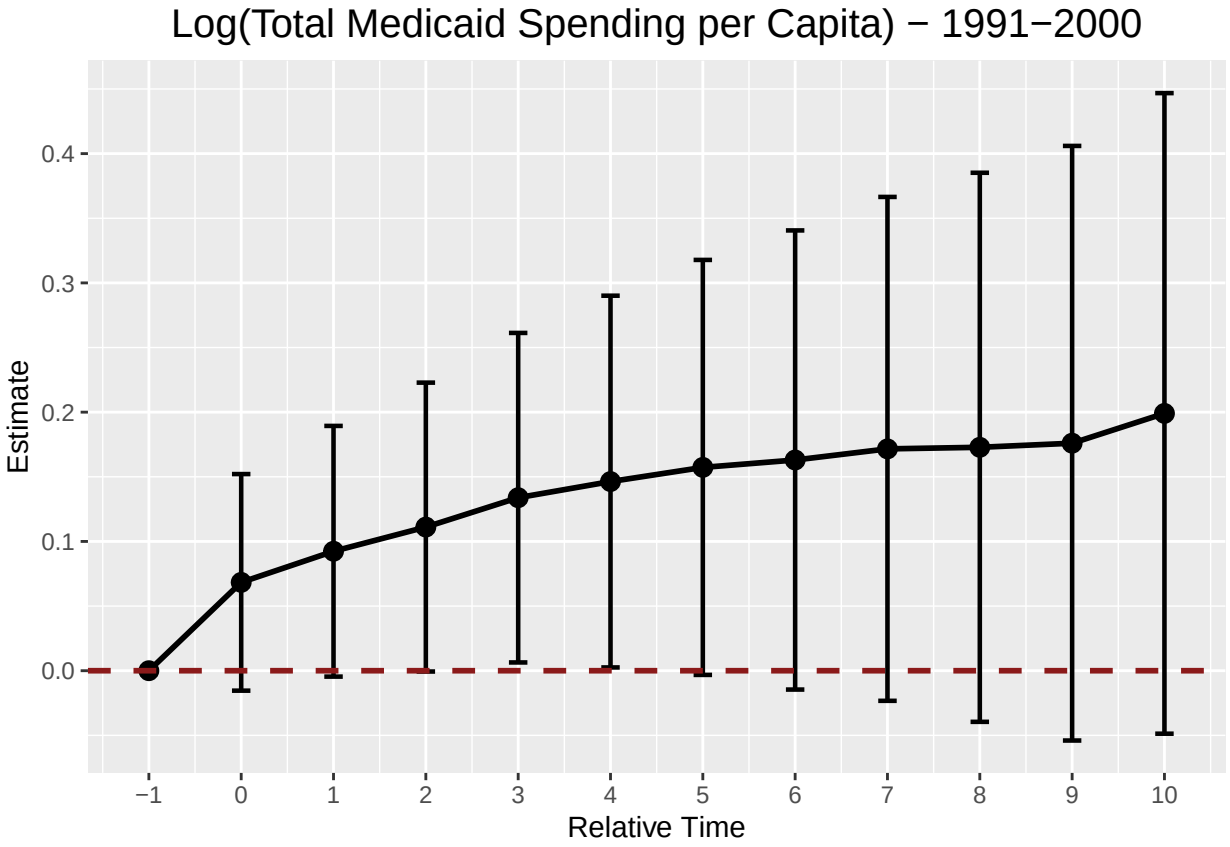

```
##
## $`log_total_spend_2011-2022`[[2]]
```



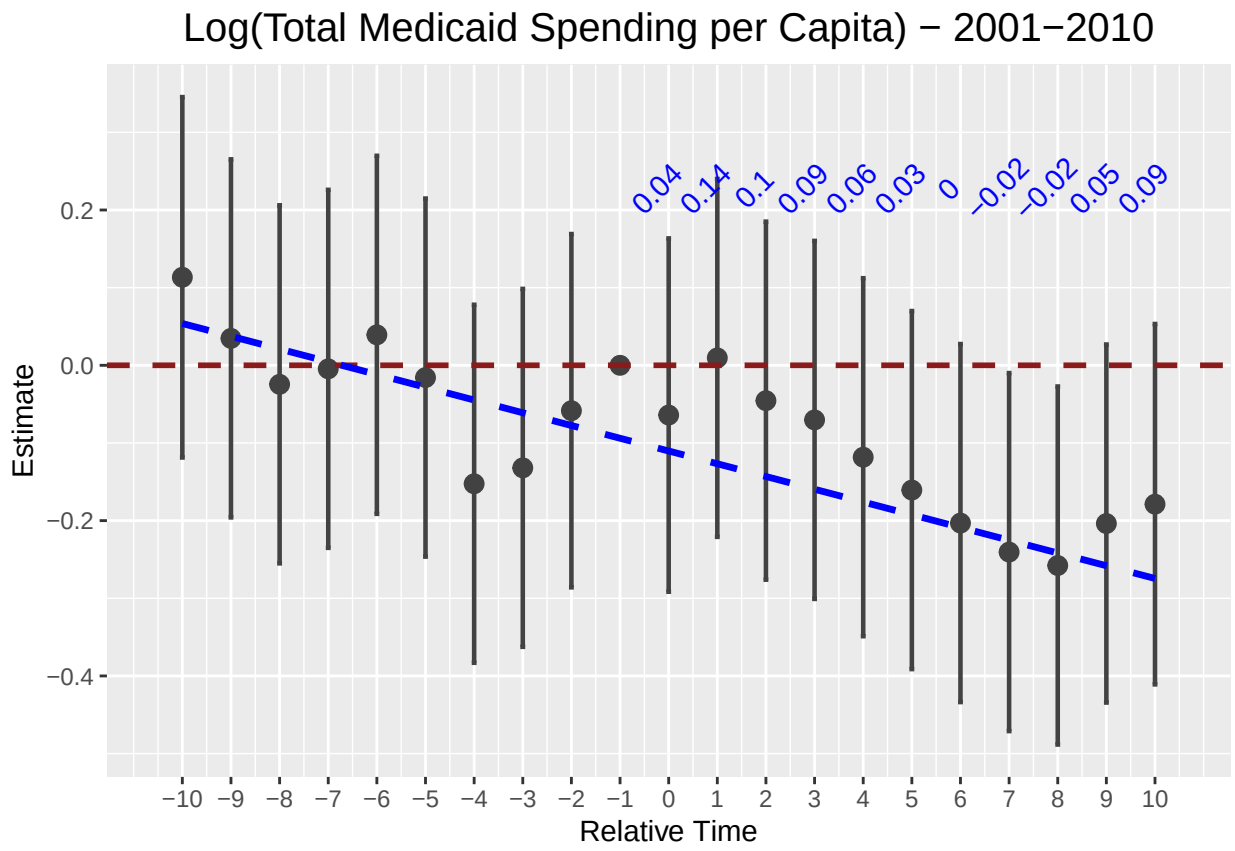

```
##
##
## $`log_all_medicaid_spending_per_cap_1991-2000`
## $`log_all_medicaid_spending_per_cap_1991-2000`[[1]]
```



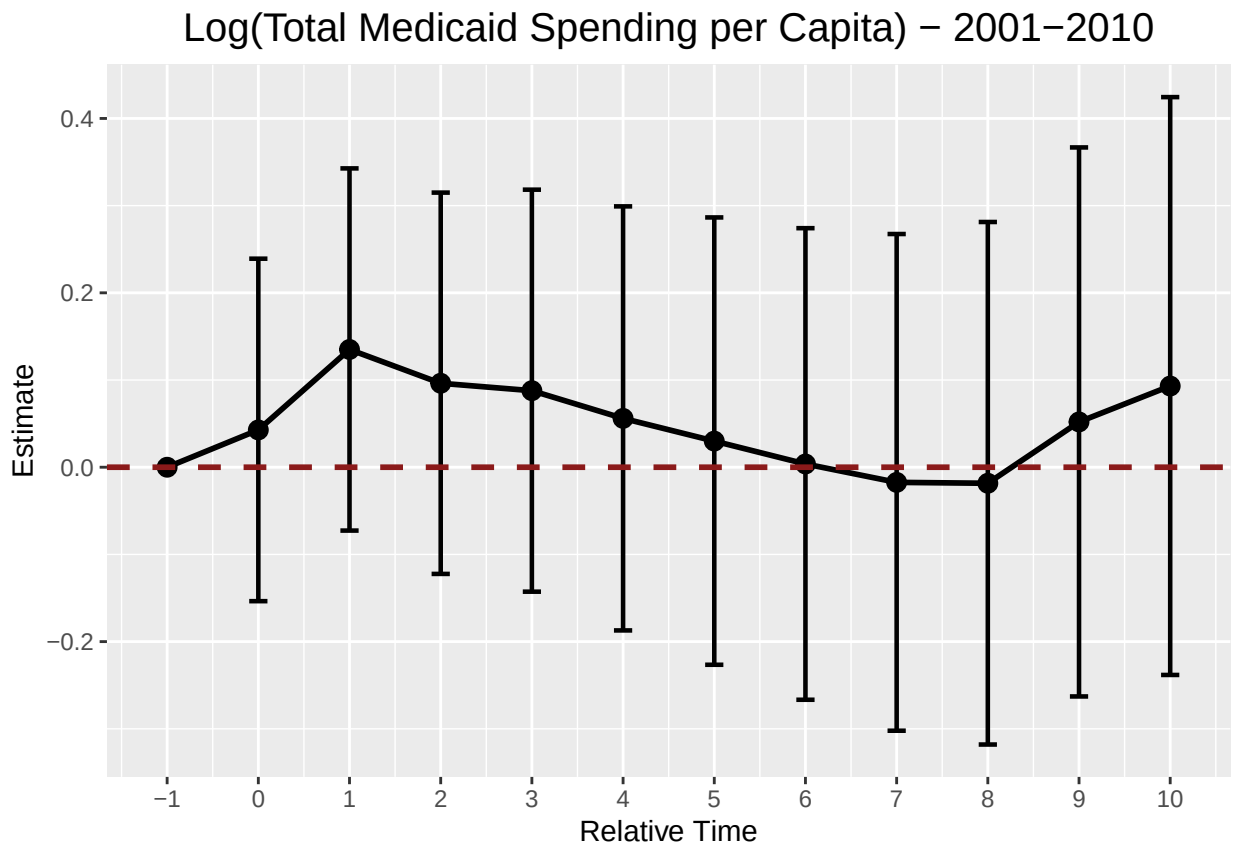

```
##
## $`log_all_medicaid_spending_per_cap_1991-2000`[[2]]
```



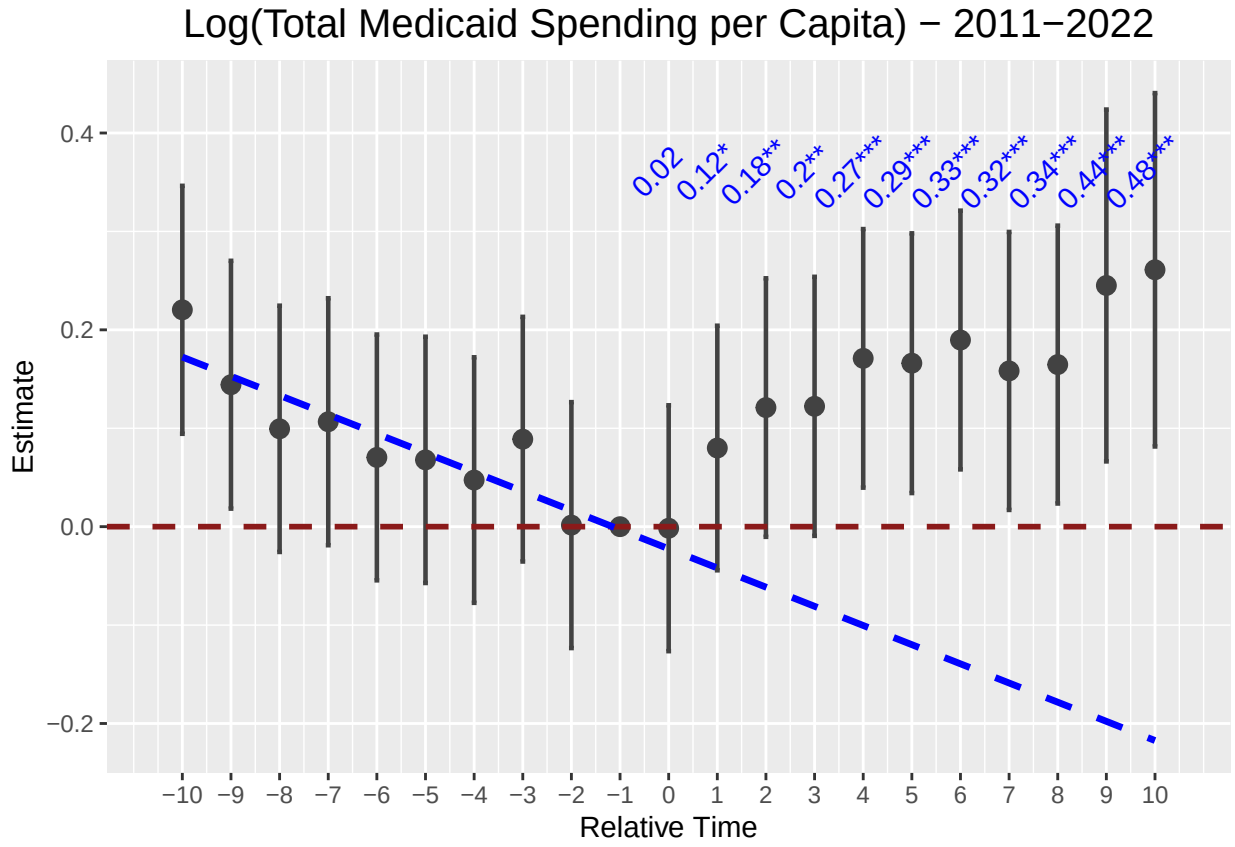

```
##
##
## $`log_all_medicaid_spending_per_cap_2001-2010`
## $`log_all_medicaid_spending_per_cap_2001-2010`[[1]]
```



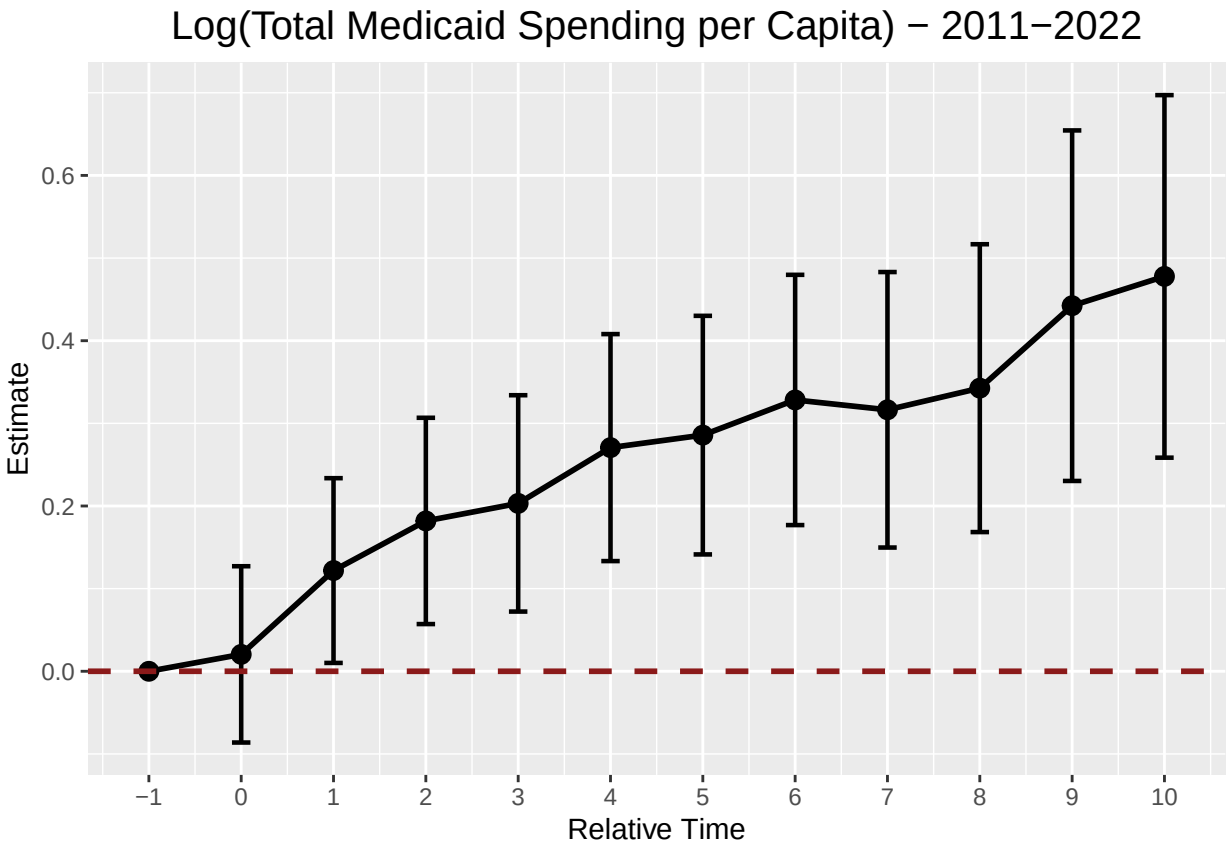

```
##
## $`log_all_medicaid_spending_per_cap_2001-2010`[[2]]
```




```
##
##
## $`log_all_medicaid_spending_per_cap_2011-2022`
## $`log_all_medicaid_spending_per_cap_2011-2022`[[1]]
```



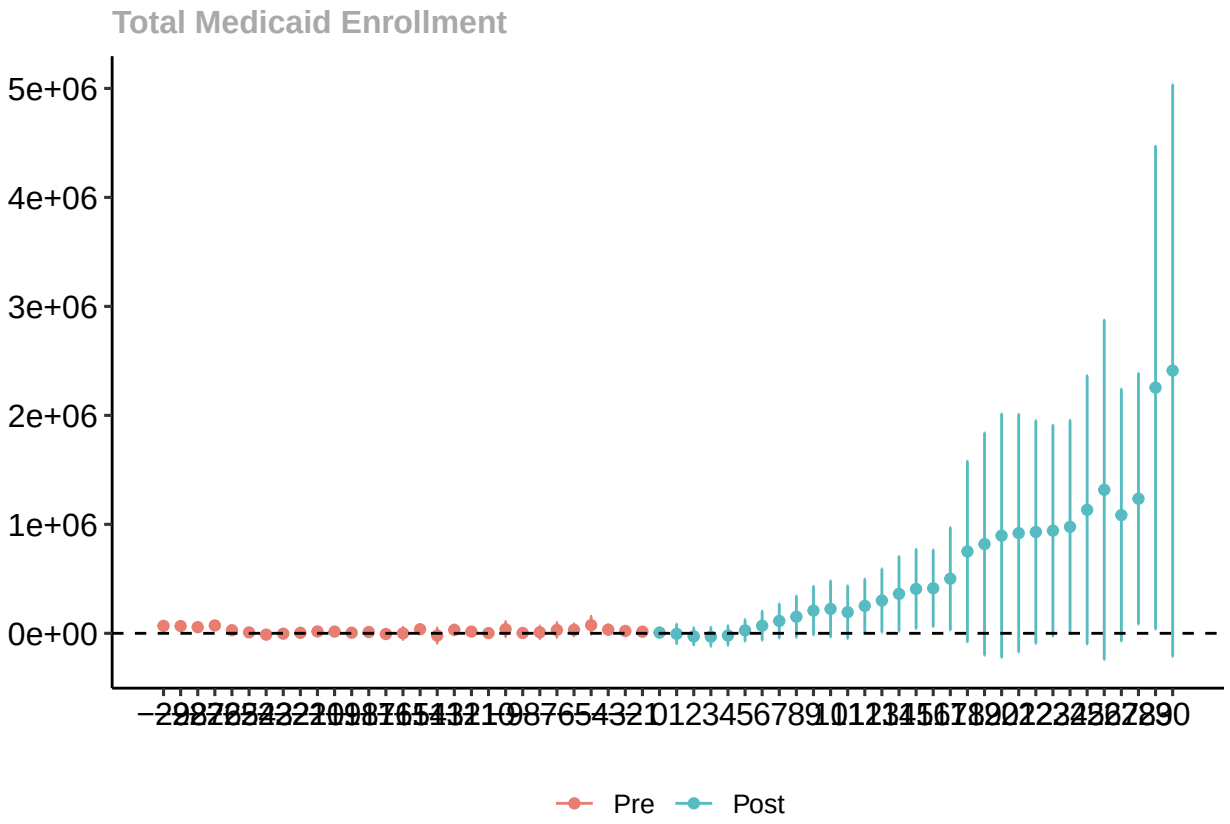

```
##
## $`log_all_medicaid_spending_per_cap_2011-2022`[[2]]
```


Log(Total Medicaid Spending per Capita) − 2011−2022
Estimate
0.6
0.4
0.2
0.0
−1
0
1
2
3
4
5
6
7
8
9
10
Relative Time

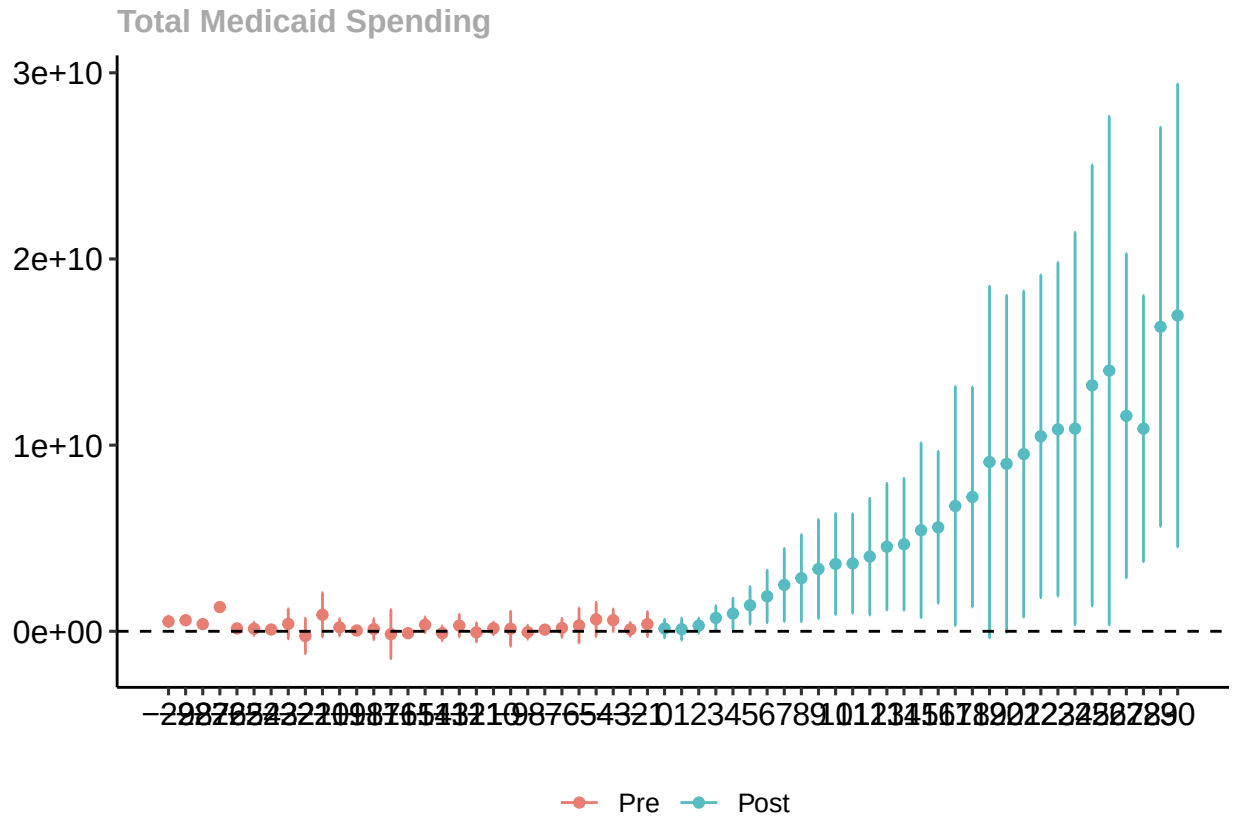
De Chaisemartin and D'Haultfoeuille (2020) (Callaway-Sant'anna)

Percent in Comprehensive MCO

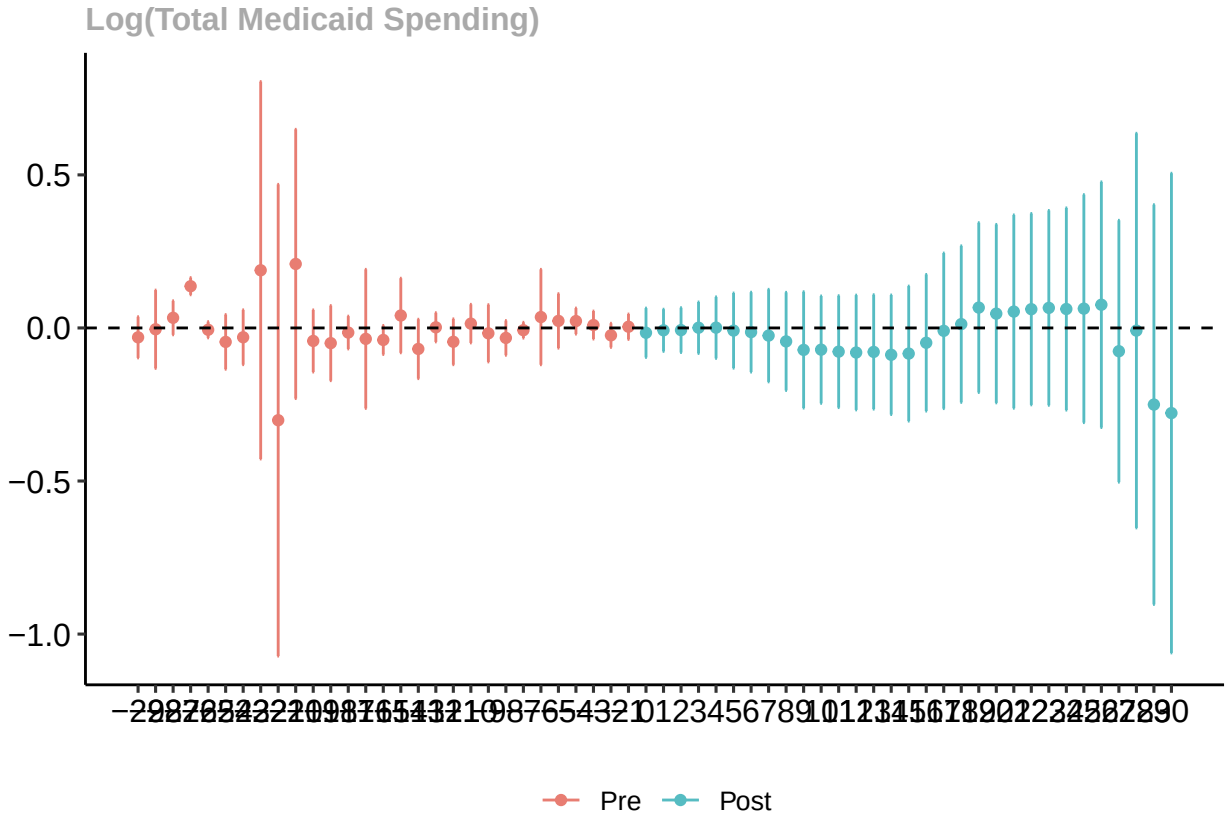
Total Medicaid Enrollment



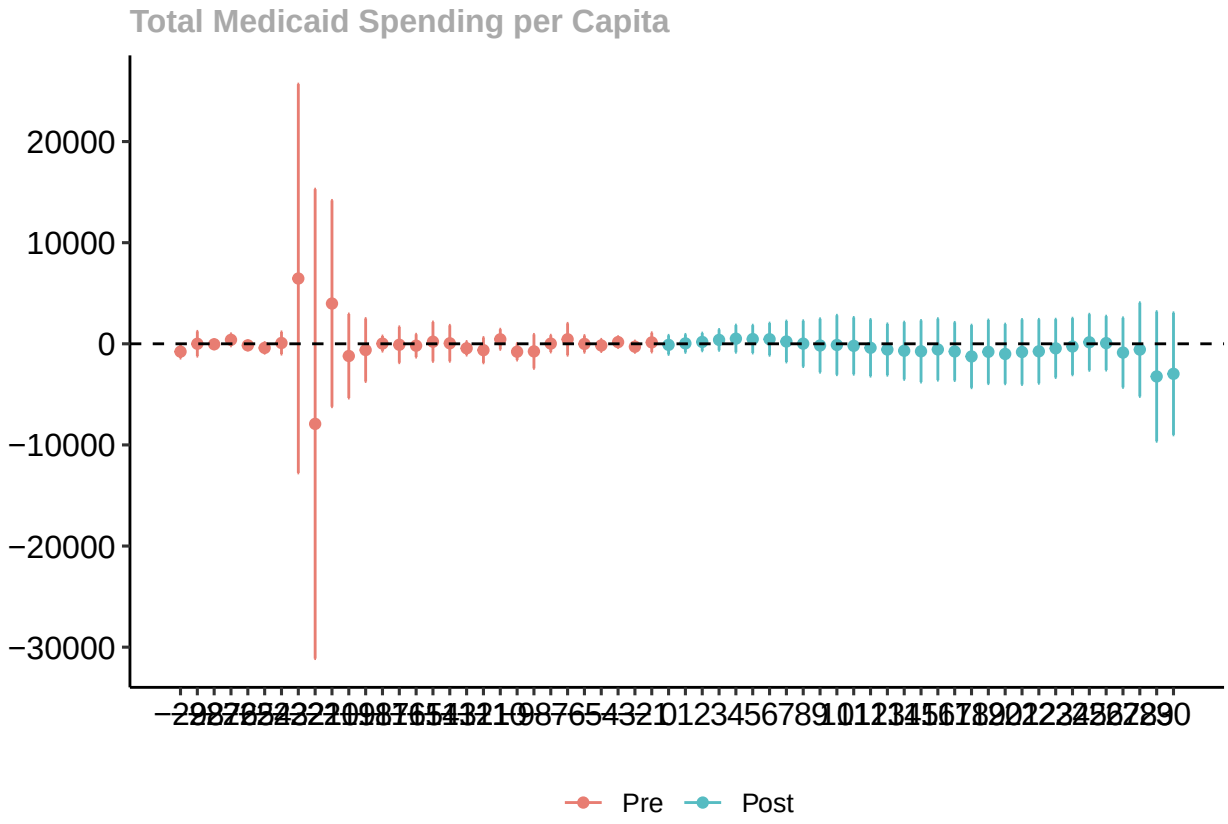
Total Medicaid Spending



Log(Total Medicaid Spending)



Total Medicaid Spending per Capita



Log(Total Medicaid Spending per Capita)