Replication Replication to Political Science Research & Methods: Rewards or Punishment? The Distribution of Life-cycle Returns to Political Office

Jens Olav Dahlgaard Frederik K. Kjøller Nicolai Kristensen

2024-09-13

Setting up pathways

```
library(tidyverse); library(haven); library(fst); library(lfe); library(lubridate); library(patchwork)
library(qte); library(broom); library(estimatr); library(ggridges); library(stargazer)

# avoid scientific notation
options(scipen = 999)

# define file paths
raw_data      <- "e:/rawdata/706687/"
work_data <- "e:/workdata/706687/"

# settings for code chunks
knitr::opts_chunk$set(warning = FALSE, message = FALSE, echo = TRUE)

# create "not in"-operator
'%!in%' <- function(x,y)!('%in%'(x,y))
```

Loading and wrangling candidate data

```
# Find everyone ever running ----------------------------------------------

candidates_2019 <-
  read_sas(paste(raw_data, "grunddata/valgdata/", "fv2019", "_recodes_pnr_afid.sas7bdat", sep = ""))

candidates_2015 <-
  read_sas(paste(raw_data, "grunddata/valgdata/", "fv2015", "_recodes_pnr_afid.sas7bdat", sep = ""))

candidates_2011 <-
  read_sas(paste(raw_data, "grunddata/valgdata/", "fv2011", "_recodes_pnr_afid.sas7bdat", sep = ""))

candidates_2007 <-
  read_sas(paste(raw_data, "grunddata/valgdata/", "fv2007", "_recodes_pnr_afid.sas7bdat", sep = ""))

candidates_2005 <-
  read_sas(paste(raw_data, "grunddata/valgdata/", "fv2005", "_recodes_pnr_afid.sas7bdat", sep = ""))
```

```
candidates_2001 <-
  read_sas(paste(raw_data, "grunddata/valgdata/", "fv2001", "_recodes_pnr_afid.sas7bdat", sep = ""))

candidates_1998 <-
  read_sas(paste(raw_data, "grunddata/valgdata/", "fv1998", "_recodes_pnr_afid.sas7bdat", sep = ""))

candidates_1994 <-
  read_sas(paste(raw_data, "grunddata/valgdata/", "fv1994", "_recodes_pnr_afid.sas7bdat", sep = ""))

candidates_1990 <-
  read_sas(paste(raw_data, "grunddata/valgdata/", "fv1990", "_recodes_pnr_afid.sas7bdat", sep = ""))

## Create panel data of all candidates - each rows contains a unique run of a candidate
all_candidates <- rbind(candidates_1990,
                        candidates_1994,
                        candidates_1998,
                        candidates_2001,
                        candidates_2005,
                        candidates_2007,
                        candidates_2011,
                        candidates_2015,
                        candidates_2019)

#### Create election variables

# create elected dummy, numeric
all_candidates <- all_candidates %>%
  dplyr::mutate(elected = case_when(VALGT_JN == "J"~1,
                                    VALGT_JN == "N"~0))

# create elected dummy as factor elected (handy for plotting)
all_candidates <- all_candidates %>%
  dplyr::mutate(factor_elected = case_when(elected==1~"Yes", elected==0~"No"))

# create variable for election year
all_candidates$election_year <-
  as.numeric(sapply(str_split(all_candidates$VALGDATO, "-"), `[`,3))

# change dae-format for election date
all_candidates <- all_candidates %>%
  mutate(election_date = dmy(VALGDATO))

# dummy for incumbency and dummy for electoral success in the next election
all_candidates <- all_candidates %>%
  arrange(PNR, election_year) %>%
  group_by(PNR) %>%
  dplyr::mutate(
    elected_lag1 = dplyr::lag(elected,1), # if NA, the candidate did not run in the previous election
    elected_lead1 = dplyr::lead(elected,1), # if NA, the candidate did not run in the next election
    elected_lead2 = dplyr::lead(elected,2),
    elected_lead3 = dplyr::lead(elected,3),
    elected_lead4 = dplyr::lead(elected,4),
    elected_lead5 = dplyr::lead(elected,5),
```

```
    elected_lead6 = dplyr::lead(elected,6),
    elected_lead7 = dplyr::lead(elected,7),
    elected_lead8 = dplyr::lead(elected,8),

    # election year - used together with next_election variables to find future incumbents
    election_year_lag1 = dplyr::lag(election_year,1),
    election_year_lead1 = dplyr::lead(election_year,1),
    election_year_lead2 = dplyr::lead(election_year,2),
    election_year_lead3 = dplyr::lead(election_year,3),
    election_year_lead4 = dplyr::lead(election_year,4),
    election_year_lead5 = dplyr::lead(election_year,5),
    election_year_lead6 = dplyr::lead(election_year,6),
    election_year_lead7 = dplyr::lead(election_year,7),
    election_year_lead8 = dplyr::lead(election_year,8))

# create dummies for first and last election
all_candidates <- all_candidates %>%
  dplyr::mutate(first_run = ifelse(is.na(elected_lag1),1,0),
                last_run = ifelse(is.na(elected_lead1),1,0)) %>%

  # assigning NA to elected lead and lag so it forms a dummy for future elected
  dplyr::mutate(elected_lag1 = replace_na(elected_lag1,0),
                elected_lead1 = replace_na(elected_lead1,0))

# create variable for future election year (needed for to matching to find future incumbents)
all_candidates <- all_candidates %>%
  dplyr::mutate(prev_election = case_when(election_year==1994~1990,
                                          election_year==1998~1994,
                                          election_year==2001~1998,
                                          election_year==2005~2001,
                                          election_year==2007~2005,
                                          election_year==2011~2007,
                                          election_year==2015~2011,
                                          election_year==2019~2015),

                next_election = case_when(election_year==1990~1994,
                                          election_year==1994~1998,
                                          election_year==1998~2001,
                                          election_year==2001~2005,
                                          election_year==2005~2007,
                                          election_year==2007~2011,
                                          election_year==2011~2015,
                                          election_year==2015~2019),

                next_2_election = case_when(election_year==1990~1998,
                                            election_year==1994~2001,
                                            election_year==1998~2005,
                                            election_year==2001~2007,
                                            election_year==2005~2011,
                                            election_year==2007~2015,
                                            election_year==2011~2019),

                next_3_election = case_when(election_year==1990~2001,
```

```
                                        election_year==1994~2005,
                                        election_year==1998~2007,
                                        election_year==2001~2011,
                                        election_year==2005~2015,
                                        election_year==2007~2019),

                next_4_election = case_when(election_year==1990~2005,
                                        election_year==1994~2007,
                                        election_year==1998~2011,
                                        election_year==2001~2015,
                                        election_year==2005~2019),

                next_5_election = case_when(election_year==1990~2007,
                                        election_year==1994~2011,
                                        election_year==1998~2015,
                                        election_year==2001~2019),

                next_6_election = case_when(election_year==1990~2011,
                                        election_year==1994~2015,
                                        election_year==1998~2019),

                next_7_election = case_when(election_year==1990~2015,
                                        election_year==1994~2019),

                next_8_election = case_when(election_year==1990~2019))


# make dummy for rerunning and incumbent for all future elections (to add allowance later)
all_candidates <- all_candidates %>%
  dplyr::mutate(running_last_elec = ifelse(election_year_lag1==prev_election,1,0),
                running_next_elec = ifelse(election_year_lead1==next_election,1,0),
                running_next_elec_2 = ifelse(election_year_lead2==next_2_election,1,0),
                running_next_elec_3 = ifelse(election_year_lead3==next_3_election,1,0),
                running_next_elec_4 = ifelse(election_year_lead4==next_4_election,1,0),
                running_next_elec_5 = ifelse(election_year_lead5==next_5_election,1,0),
                running_next_elec_6 = ifelse(election_year_lead6==next_6_election,1,0),
                running_next_elec_7 = ifelse(election_year_lead7==next_7_election,1,0),
                running_next_elec_8 = ifelse(election_year_lead8==next_8_election,1,0)) %>%

  dplyr::mutate(incumbent = ifelse(running_last_elec==1 & elected_lag1==1,1,0),
                incumbent_future = ifelse(running_next_elec==1 & elected_lead1==1,1,0),
                incumbent_future_2 = ifelse(running_next_elec_2==1 & elected_lead2==1,1,0),
                incumbent_future_3 = ifelse(running_next_elec_3==1 & elected_lead3==1,1,0),
                incumbent_future_4 = ifelse(running_next_elec_4==1 & elected_lead4==1,1,0),
                incumbent_future_5 = ifelse(running_next_elec_5==1 & elected_lead5==1,1,0),
                incumbent_future_6 = ifelse(running_next_elec_6==1 & elected_lead6==1,1,0),
                incumbent_future_7 = ifelse(running_next_elec_7==1 & elected_lead7==1,1,0),
                incumbent_future_8 = ifelse(running_next_elec_8==1 & elected_lead8==1,1,0)) %>%

  # assigning 0 to all NAs (NAs occur for years where the candidate don't run)
  dplyr::mutate(running_last_elec = replace_na(running_last_elec,0),
                running_next_elec = replace_na(running_next_elec,0),
                running_next_elec_2 = replace_na(running_next_elec_2,0),
```

```
                running_next_elec_3 = replace_na(running_next_elec_3,0),
                running_next_elec_4 = replace_na(running_next_elec_4,0),
                running_next_elec_5 = replace_na(running_next_elec_5,0),
                running_next_elec_6 = replace_na(running_next_elec_6,0),
                running_next_elec_7 = replace_na(running_next_elec_7,0),
                running_next_elec_8 = replace_na(running_next_elec_8,0),
                incumbent = replace_na(incumbent,0),
                incumbent_future = replace_na(incumbent_future,0),
                incumbent_future_2 = replace_na(incumbent_future_2,0),
                incumbent_future_3 = replace_na(incumbent_future_3,0),
                incumbent_future_4 = replace_na(incumbent_future_4,0),
                incumbent_future_5 = replace_na(incumbent_future_5,0),
                incumbent_future_6 = replace_na(incumbent_future_6,0),
                incumbent_future_7 = replace_na(incumbent_future_7,0),
                incumbent_future_8 = replace_na(incumbent_future_8,0))

## create variable for political party
# make a variable based on the first letter in party variable.
all_candidates <- all_candidates %>%
  dplyr::mutate(party = substring(PARTI,1,1))

# create factor variable for political party
all_candidates$party_factor <- as.factor(all_candidates$party)
levels(all_candidates$party_factor) # 20 parties
```

```
##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "K" "M" "O" "P" "Q" "U" "V" "Y" "Z" "Ø"
## [20] "Å"
```

```
# name parties in a factor variable
# note that after the election of 2019 two new parties have got former parties' letters:
# P, Stram Kurs, D for Nye Borgerlige, E for Klaus Riskær -

all_cand_before_2015 <- all_candidates %>%
  filter(election_year<2016)

all_cand_before_2015$party_factor <- all_cand_before_2015$party_factor %>%
  fct_collapse(socialdemocrats = "A",
               socialliberals = "B",
               conservatives = "C",
               centrumdemocrats = "D",
               justiceparty = "E",
               socialistpeoplesparty = "F",
               thegreens = "G",
               humanistparty = "H",
               liberalalliance = c("I", "Y"),
               christiandemocrats = c("K", "Q"),
               minorityparty = "M",
               danishpeoplesparty = "O",
               commoncourse = "P",
               democraticrenewal = "U",
               liberals = "V",
               progressparty = "Z",
               redgreenalliance = "Ø",
               thealternative = "Å")
```

```r
all_cand_after_2015 <- all_candidates %>%
  filter(election_year>2016)

all_cand_after_2015$party_factor <- all_cand_after_2015$party_factor %>%
  fct_collapse(socialdemocrats = "A",
               socialliberals = "B",
               conservatives = "C",
               newright = "D",
               klausriskaer = "E",
               socialistpeoplesparty = "F",
               liberalalliance = "I",
               christiandemocrats = "K",
               danishpeoplesparty = "O",
               hardline = "P",
               outsideparties = "U",
               liberals = "V",
               redgreenalliance = "Ø",
               thealternative = "Å")

all_candidates <- rbind(all_cand_before_2015, all_cand_after_2015)

all_candidates <- all_candidates %>%
  dplyr::mutate(rightwing = case_when(party_factor=="socialdemocrats"~0,
                                      party_factor=="socialliberals" & election_year<1994~1,
                                      party_factor=="socialliberals" & election_year>=1994~0,
                                      party_factor=="conservatives"~1,
                                      party_factor=="centrumdemocrats" & election_year<1994~1,
                                      party_factor=="centrumdemocrats" & election_year>=1994~0,
                                      party_factor=="justiceparty"~0,
                                      party_factor=="socialistpeoplesparty"~0,
                                      party_factor=="thegreens"~0,
                                      party_factor=="humanistparty"~0,
                                      party_factor=="liberalalliance"~1,
                                      party_factor=="christiandemocrats" & election_year<1994~1,
                                      party_factor=="christiandemocrats" & election_year>=1994~0,
                                      party_factor=="minorityparty"~0,
                                      party_factor=="danishpeoplesparty"~1,
                                      party_factor=="commoncourse"~0,
                                      party_factor=="democraticrenewal"~0,
                                      party_factor=="liberals"~1,
                                      party_factor=="progressparty"~1,
                                      party_factor=="redgreenalliance"~0,
                                      party_factor=="thealternative"~0,
                                      party_factor=="newright"~1,
                                      party_factor=="klausriskaer"~1,
                                      party_factor=="hardline"~1,
                                      party_factor=="outsideparties"~0))

all_candidates <- all_candidates %>%
  mutate(majority_coalition = case_when(election_year==1990 & party_factor=="socialliberals"~1,
                                        election_year==1990 & party_factor=="conservatives"~1,
                                        election_year==1990 & party_factor=="centrumdemocrats"~1,
                                        election_year==1990 & party_factor=="christiandemocrats"~1,
```

```
                                    election_year==1990 & party_factor=="liberals"~1,
                                    election_year==1990 & party_factor=="progressparty"~1,

                                    election_year==1994 & party_factor=="socialdemocrats"~1,
                                    election_year==1994 & party_factor=="socialliberals"~1,
                                    election_year==1994 & party_factor=="socialistpeoplesparty"~1,
                                    election_year==1994 & party_factor=="centrumdemocrats"~1,
                                    election_year==1994 & party_factor=="redgreenalliance"~1,

                                    election_year==1998 & party_factor=="socialdemocrats"~1,
                                    election_year==1998 & party_factor=="socialliberals"~1,
                                    election_year==1998 & party_factor=="socialistpeoplesparty"~1,
                                    election_year==1998 & party_factor=="redgreenalliance"~1,

                                    election_year==2001 & party_factor=="conservatives"~1,
                                    election_year==2001 & party_factor=="liberals"~1,
                                    election_year==2001 & party_factor=="danishpeoplesparty"~1,

                                    election_year==2005 & party_factor=="conservatives"~1,
                                    election_year==2005 & party_factor=="liberals"~1,
                                    election_year==2005 & party_factor=="danishpeoplesparty"~1,

                                    election_year==2007 & party_factor=="conservatives"~1,
                                    election_year==2007 & party_factor=="liberals"~1,
                                    election_year==2007 & party_factor=="danishpeoplesparty"~1,

                                    election_year==2011 & party_factor=="socialdemocrats"~1,
                                    election_year==2011 & party_factor=="socialliberals"~1,
                                    election_year==2011 & party_factor=="socialistpeoplesparty"~1,
                                    election_year==2011 & party_factor=="redgreenalliance"~1,

                                    election_year==2015 & party_factor=="conservatives"~1,
                                    election_year==2015 & party_factor=="liberals"~1,
                                    election_year==2015 & party_factor=="danishpeoplesparty"~1,
                                    election_year==2015 & party_factor=="liberalalliance"~1,

                                    election_year==2019 & party_factor=="socialdemocrats"~1,
                                    election_year==2019 & party_factor=="socialliberals"~1,
                                    election_year==2019 & party_factor=="socialistpeoplesparty"~1,
                                    election_year==2019 & party_factor=="redgreenalliance"~1,

                                    TRUE~0))


all_candidates <- all_candidates %>%
  as.data.frame()

# create clusterid
# cluster id for party x district (storkreds) x year

all_candidates$cluster_id <- all_candidates %>%
  group_by(PARTI, STORKREDS, election_year) %>%
  group_indices()
```

```r
all_candidates$cluster_id <- as.factor(all_candidates$cluster_id)


# create a list where the candidate pool (dataframe) of each election is an element
candidates_list <- split(all_candidates, all_candidates$election_year)

# name each element (election dataframe) in the list
names(candidates_list) <- c("candidates_1990",
                            "candidates_1994",
                            "candidates_1998",
                            "candidates_2001",
                            "candidates_2005",
                            "candidates_2007",
                            "candidates_2011",
                            "candidates_2015",
                            "candidates_2019")

save(candidates_list,
     file = paste0(work_data, "projects/rewards_or_punishment/replication/candidates_list.rdata"))
```

Data wrangling

```r
load(paste0(work_data, "projects/rewards_or_punishment/data/data_list.rdata"))


load(paste0(work_data, "projects/rewards_or_punishment/replication/candidates_list.rdata"))
list2env(candidates_list ,.GlobalEnv)
```

```
## <environment: R_GlobalEnv>
```

```r
# subset to only freshmen/first time runners and candidates from open-list parties (sideordnet)
# note that for every candidate in 1990 first_run==1
for (i in 1:length(candidates_list)){
  candidates_list[[i]] <- subset(candidates_list[[i]], first_run==1)
  candidates_list[[i]] <- subset(candidates_list[[i]], OPSTILLINGSFORM=="Sideordnet")
}

list2env(candidates_list ,.GlobalEnv)
```

```
## <environment: R_GlobalEnv>
```

```r
rm(candidates_2019) # remove 2019 candidates -- we have no outcome years anyways...

# joining lists of data - include administrative data for each candidate in
# t-6 to 2018 (latest available income year) for each election
election_1990_list <- list()
for (year in 1987:2018) {
  election_1990_list[[as.character(year)]] <- data_list[[as.character(year)]] %>%
    filter(PNR %in% candidates_1990$PNR) %>%
    left_join(candidates_1990, by = "PNR")
}
```

```
election_1994_list <- list()
for (year in 1988:2018) {
  election_1994_list[[as.character(year)]] <- data_list[[as.character(year)]] %>%
    filter(PNR %in% candidates_1994$PNR) %>%
    left_join(candidates_1994, by = "PNR")
}

election_1998_list <- list()
for (year in 1992:2018) {
  election_1998_list[[as.character(year)]] <- data_list[[as.character(year)]] %>%
    filter(PNR %in% candidates_1998$PNR) %>%
    left_join(candidates_1998, by = "PNR")
}

election_2001_list <- list()
for (year in 1995:2018) {
  election_2001_list[[as.character(year)]] <- data_list[[as.character(year)]] %>%
    filter(PNR %in% candidates_2001$PNR) %>%
    left_join(candidates_2001, by = "PNR")
}

election_2005_list <- list()
for (year in 1999:2018) {
  election_2005_list[[as.character(year)]] <- data_list[[as.character(year)]] %>%
    filter(PNR %in% candidates_2005$PNR) %>%
    left_join(candidates_2005, by = "PNR")
}

election_2007_list <- list()
for (year in 2001:2018) {
  election_2007_list[[as.character(year)]] <- data_list[[as.character(year)]] %>%
    filter(PNR %in% candidates_2007$PNR) %>%
    left_join(candidates_2007, by = "PNR")
}

election_2011_list <- list()
for (year in 2005:2018) {
  election_2011_list[[as.character(year)]] <- data_list[[as.character(year)]] %>%
    filter(PNR %in% candidates_2011$PNR) %>%
    left_join(candidates_2011, by = "PNR")
}

election_2015_list <- list()
for (year in 2009:2018) {
  election_2015_list[[as.character(year)]] <- data_list[[as.character(year)]] %>%
    filter(PNR %in% candidates_2015$PNR) %>%
    left_join(candidates_2015, by = "PNR")
}

election_1990 <- do.call(bind_rows, election_1990_list)
election_1994 <- do.call(bind_rows, election_1994_list)
election_1998 <- do.call(bind_rows, election_1998_list)
election_2001 <- do.call(bind_rows, election_2001_list)
```

```r
election_2005 <- do.call(bind_rows, election_2005_list)
election_2007 <- do.call(bind_rows, election_2007_list)
election_2011 <- do.call(bind_rows, election_2011_list)
election_2015 <- do.call(bind_rows, election_2015_list)

full_elections <- list(election_1990,
                       election_1994,
                       election_1998,
                       election_2001,
                       election_2005,
                       election_2007,
                       election_2011,
                       election_2015)

#### Data manipulation ----------

# creates df for the sample - long format
df <- do.call(bind_rows, full_elections)

#create t-variable
df <- df %>%
  arrange(PNR,year) %>%
  group_by(PNR) %>%
  dplyr::mutate(t = year-election_year) %>%
  ungroup()

# defining treatment-period and did
df <- df %>%
  dplyr::mutate(post_election = ifelse(year>election_year & year<next_election,1,0),
                did = elected*post_election)

# defining placebo did
df <- df %>%
  dplyr::mutate(placebo_post_m2 = ifelse(election_year-2 <= year,1,0),
                placebo_post_m3 = ifelse(election_year-3 <= year,1,0),

                placebo_did_m2 = elected*placebo_post_m2,
                placebo_did_m3 = elected*placebo_post_m3)


# for candidates in 1998 add avg. employer paid pension of 1995-1997 to 1994
df_pens_1994 <- df %>%
  filter(election_year==1998) %>%
  filter(year>1994 & year < 1998) %>% # only include 1995-1997
  dplyr::mutate(pension_employer = replace_na(pension_employer,0),
                pension_employer_euro = replace_na(pension_employer_euro,0)) %>%
  group_by(PNR) %>%
  summarise(mean_pre_pension = mean(pension_employer),
            mean_pre_pension_euro = mean(pension_employer_euro))

# add to df
df <- df %>%
  left_join(df_pens_1994, by = "PNR")
```

```r
# add mean of 1995-1997 to 1994 for candidates of 1998
df <- df %>%
  dplyr::mutate(pension_employer =
                  ifelse(election_year==1998 & year==1994, mean_pre_pension, pension_employer))

# now calculate average employer paid pension avg. from 1994-1997
# and assign to 1994-candidates observations of 1990-1993
avg_pension <- df %>%
  filter(election_year==1998) %>%
  filter(year>1993 & year < 1998) %>% # only include pre-treatment
  dplyr::mutate(pension_employer = replace_na(pension_employer,0),
                pension_employer_euro = replace_na(pension_employer_euro,0)) %>%
  summarise(mean_pre_pension = mean(pension_employer),
            mean_pre_pension_euro = mean(pension_employer_euro))


# assign to 1994-candidates
df <- df %>%
  mutate(pension_employer =
         ifelse(election_year==1994, avg_pension$mean_pre_pension, pension_employer),
         pension_employer_euro =
         ifelse(election_year==1994, avg_pension$mean_pre_pension_euro, pension_employer_euro ))

# assign 0 to pension NA's:
df <- df %>%
  dplyr::mutate(pension_employer = replace_na(pension_employer,0),
                pension_employer_euro = replace_na(pension_employer_euro,0))

#### Add tax-extempted allowance
df <- df %>%
  dplyr::mutate(untouched_income = total_income_euro) %>%
  dplyr::mutate(total_income_pension_euro = total_income_euro + pension_employer_euro) %>%
  dplyr::mutate(#1st election period
    total_income_euro = ifelse(did==1, total_income_euro+18365,total_income_euro),
    income_m_taxes_euro = ifelse(did==1, income_m_taxes_euro+8057,income_m_taxes_euro),
    total_income_pension_euro = ifelse(did==1, total_income_euro+27346,total_income_euro)) %>%

  # 2nd election period after first run
  dplyr::mutate(
    total_income_euro = ifelse(year>=next_election & year<next_2_election & incumbent_future_2==1,
                               total_income_euro+18365,total_income_euro),
    income_m_taxes_euro = ifelse(year>=next_election & year<next_2_election & incumbent_future_2==1,
                                 income_m_taxes_euro+8057, income_m_taxes_euro),
    total_income_pension_euro =
      ifelse(year>=next_election & year<next_2_election & incumbent_future_2==1,
                                      total_income_pension_euro+27346,total_income_pension_euro)) %>%
  dplyr::mutate(
    # 3rd election period after first run
    total_income_euro = ifelse(year>=next_2_election & year<next_3_election & incumbent_future_3==1,
                               total_income_euro+18365,total_income_euro),
    income_m_taxes_euro = ifelse(year>=next_2_election & year<next_3_election & incumbent_future_3==1,
                                 income_m_taxes_euro+8057, income_m_taxes_euro),
    total_income_pension_euro =
```

```
    ifelse(year>=next_2_election & year<next_3_election & incumbent_future_3==1,
                                    total_income_pension_euro+27346,total_income_pension_euro)) %>%

dplyr::mutate(
  # 4th election period after first run
  total_income_euro = ifelse(year>=next_3_election & year<next_4_election & incumbent_future_4==1,
                             total_income_euro+18365,total_income_euro),
  income_m_taxes_euro = ifelse(year>=next_3_election & year<next_4_election & incumbent_future_4==1,
                               income_m_taxes_euro+8057, income_m_taxes_euro),
  total_income_pension_euro =
    ifelse(year>=next_3_election & year<next_4_election & incumbent_future_4==1,
                                    total_income_pension_euro+27346,total_income_pension_euro)) %>%
dplyr::mutate(
  # 5th election period after first run
  total_income_euro = ifelse(year>=next_4_election & year<next_5_election & incumbent_future_5==1,
                             total_income_euro+18365,total_income_euro),
  income_m_taxes_euro = ifelse(year>=next_4_election & year<next_5_election & incumbent_future_5==1,
                               income_m_taxes_euro+8057, income_m_taxes_euro),
  total_income_pension_euro =
    ifelse(year>=next_4_election & year<next_5_election & incumbent_future_5==1,
                                    total_income_pension_euro+27346,total_income_pension_euro)) %>%
dplyr::mutate(
  # 6th election period after first run
  total_income_euro = ifelse(year>=next_5_election & year<next_6_election & incumbent_future_6==1,
                             total_income_euro+18365,total_income_euro),
  income_m_taxes_euro = ifelse(year>=next_5_election & year<next_6_election & incumbent_future_6==1,
                               income_m_taxes_euro+8057, income_m_taxes_euro),
  total_income_pension_euro =
    ifelse(year>=next_5_election & year<next_6_election & incumbent_future_6==1,
                                    total_income_pension_euro+27346,total_income_pension_euro)) %>%
dplyr::mutate(
  # 6th election period after first run
  total_income_euro = ifelse(year>=next_6_election & year<next_7_election & incumbent_future_7==1,
                             total_income_euro+18365,total_income_euro),
  income_m_taxes_euro = ifelse(year>=next_6_election & year<next_7_election & incumbent_future_7==1,
                               income_m_taxes_euro+8057, income_m_taxes_euro),
  total_income_pension_euro =
    ifelse(year>=next_6_election & year<next_7_election & incumbent_future_7==1,
                                    total_income_pension_euro+27346,total_income_pension_euro)) %>%
dplyr::mutate(

  # 7th election period after first run
  total_income_euro = ifelse(year>=next_7_election & year<next_8_election & incumbent_future_8==1,
                             total_income_euro+18365,total_income_euro),
  income_m_taxes_euro = ifelse(year>=next_7_election & year<next_8_election & incumbent_future_8==1,
                               income_m_taxes_euro+8057, income_m_taxes_euro),
  total_income_pension_euro =
    ifelse(year>=next_7_election & year<next_8_election & incumbent_future_8==1,
                                    total_income_pension_euro+27346,total_income_pension_euro)) %>%

# add pensions (arbejdsgiver administrerede)
dplyr::mutate(total_income_pension_euro = total_income_pension_euro + pension_employer_euro)
```

```r
# removing variables used to create future incumbency dummies

df <- df %>%
  select(-c(elected_lag1, elected_lead1, elected_lead2, elected_lead3, elected_lead4, elected_lead5,
            elected_lead6, elected_lead7, elected_lead8,
            election_year_lag1, election_year_lead1, election_year_lead2, election_year_lead3,
            election_year_lead4, election_year_lead5, election_year_lead6, election_year_lead7,
            election_year_lead8,
            prev_election, next_election, next_2_election, next_3_election, next_4_election,
            next_5_election,next_6_election, next_7_election, next_8_election))

######################
#### DESCRIPTIVES ###
######################

# Creating factor and collapsing levels into 5
df$education <- factor(df$education) %>%
  fct_collapse(primaryschool = "Grundskole",
               highschool = c("Almengymnasiale uddannelser","Erhvervsgymnasiale uddannelser",
                              "Korte videregående uddannelser"),
               vocational = "Erhvervsfaglige praktik- og hovedforløb",
               mediumlong = c("Mellemlange videregående uddannelser", "Bachelor"),
               long = c("Lange videregående uddannelser","Forskeruddannelser"))

df$education <- factor(df$education,
                                levels = c("primaryschool", "highschool", "vocational",
                                           "mediumlong","long"))

df <- df %>%
  dplyr::mutate(primaryschooleducation = ifelse(education=="primaryschool",1,0),
                highschooleducation = ifelse(education=="highschool",1,0),
                vocationaleducation = ifelse(education=="vocational",1,0),
                mediumlongeducation = ifelse(education=="mediumlong",1,0),
                longeducation = ifelse(education=="long",1,0))


#various observables
df  <- df  %>%
  dplyr::mutate(female = KOEN-1,
                pnr_factor = factor(PNR),
                age2 = age^2,
                election_year_factor = factor(election_year)) %>%
  rename(civst = CIVST)

# Age groups
df_1pre <- df %>%
  filter(year==election_year-1) %>% # looking at age in t-1
  mutate(age_group = case_when(age<30~"17-29",
                               30<=age & age<40~"30-39",
                               40<=age & age<50~"40-49",
                               50<=age & age<60~"50-59",
                               60<=age~"60+")) %>%
  select(PNR, age_group)
```

```r
df <- df %>%
  left_join(df_1pre, by = "PNR") %>%
  mutate(age_group = ifelse(!is.na(age_group),age_group,case_when(age<30~"17-29",
                                                                  30<=age & age<40~"30-39",
                                                                  40<=age & age<50~"40-49",
                                                                  50<=age & age<60~"50-59",
                                                                  60<=age~"60+")))

df <- df %>%
  mutate(age_group_factor = as.factor(age_group))

# marital status
df$civst <- df$civst %>%
  fct_collapse(married = "G",
               not_married = "U",
               divorced = "F",
               another_cvist = c("P", "O", "E"))

df <- df %>%
  dplyr::mutate(married = ifelse(civst=="married",1,0))




#### Make pre-election avg. income and pre-election income quartiles

df_pre <- df %>%
  filter(t<0) %>%
  group_by(PNR) %>%
  summarise(avg_income_pension_pre = mean(total_income_pension_euro, na.rm=T),
            avg_earnings_pre = mean(untouched_income, na.rm=T)) %>%
  dplyr::mutate(income_pre_percentile = ntile(avg_income_pension_pre, 4)) %>%
  ungroup()

df <- df %>%
  left_join(df_pre, by = "PNR")

## calculate partner income
df <- df %>%
  mutate(partner_income = fam_total_income_euro-total_income_euro)


save(df,
     file = paste0(work_data, "projects/rewards_or_punishment/replication/df_sample_manipulated.rdata"))
```

Figure 1

```
# find subsample's annual estimated earnings from t-4
df_pt <- df %>%
  filter(election_year>1993) %>%
  filter(fractile_thres<5) %>% # select five percent closest elections (0,1,2,3,4)
  filter(t!=0) %>% # do not include election year
  filter(t<4) %>%
  as.data.frame()


plot_pt_naive <- df_pt %>%
  group_by(factor_elected, t) %>%
  summarise(mean_income = mean(untouched_income, na.rm=TRUE),
            std_error = sqrt(var(untouched_income)/length(untouched_income)-1)) %>%
  mutate(outcome = "Earnings")

plot_pt_allowance_pension <- df_pt %>%
  group_by(factor_elected, t) %>%
  summarise(mean_income = mean(total_income_pension_euro, na.rm=TRUE),
            std_error = sqrt(var(total_income_pension_euro)/length(total_income_pension_euro)-1)) %>%
  mutate(outcome = "Total Income")

plot_pt <- bind_rows(plot_pt_naive, plot_pt_allowance_pension) %>%
  mutate(outcome = factor(outcome, levels = c("Earnings",
                                              "Total Income")))
plot1 <-
plot_pt %>%
  filter(t<3) %>%
  ggplot(data=., aes(x=t, y=mean_income, color = factor_elected, shape = factor_elected)) +
  geom_point(position = position_dodge(width = 0.6), size = 2.5) +
  geom_linerange(position = position_dodge(width = 0.6),
                 aes(ymin=mean_income-(std_error*1.96),
                     ymax=mean_income+(std_error*1.96)),
                 size = 0.7) +
  geom_vline(xintercept = 0, linetype = "longdash") +
  scale_x_continuous(breaks = seq(-6,3,1), labels = seq(-6,3,1)) +
  scale_y_continuous(breaks = seq(0,160000,20000), labels = scales::comma) +
  xlab("") +
  ylab("Euros") +
  labs(color="Elected", shape="Elected") +
  scale_color_manual(values = c("black", "grey55")) +
  scale_alpha_manual(values = c(0.2, 0.5, 0.8)) +
  theme_bw() +
  coord_cartesian(ylim = c(0,160000)) +
  facet_wrap(~outcome)

#### Callaway sant'Anna
df_att <- df_pt %>%
  mutate(t_factor = factor(t, levels = c(-1,-3,-2, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
                                         11, 12, 13, 14, 15, 16)),
         pnr_num = as.numeric(PNR))

att_naive_reg <- did::att_gt(yname = "untouched_income",
```

```
                          tname = "t",
                          id = "pnr_num",
                          gname = "elected",
                          est_method = "dr",
                          control_group = "nevertreated",
                          anticipation = 0,
                          panel = TRUE,
                          clustervars = "cluster_id",
                          bstrap = TRUE,
                          biters = 1000,
                          base_period = "universal",
                          allow_unbalanced_panel = TRUE,
                          pl = TRUE,
                          cores = 6,
                          data = df_att)

att_naive <- data.frame(outcome = "Earnings",
                            t = att_naive_reg$t,
                            att = att_naive_reg$att,
                            se = att_naive_reg$se)


att_allow_pension_reg <- did::att_gt(yname = "total_income_pension_euro",
                          tname = "t",
                          id = "pnr_num",
                          gname = "elected",
                          est_method = "dr",
                          control_group = "nevertreated",
                          anticipation = 0,
                          panel = TRUE,
                          clustervars = "cluster_id",
                          bstrap = TRUE,
                          biters = 1000,
                          base_period = "universal",
                          allow_unbalanced_panel = TRUE,
                          pl = TRUE,
                          cores = 6,
                          data = df_att)

att_allow_pension <- data.frame(outcome = "Total Income",
                        t = att_allow_pension_reg$t,
                        att = att_allow_pension_reg$att,
                        se = att_allow_pension_reg$se)


att_plot <- bind_rows(att_naive, att_allow_pension) %>%
  mutate(outcome = factor(outcome, levels = c("Earnings", "Earnings + Allowance",
                                              "Total Income")))

plot2 <-
att_plot %>%
  filter(t<3) %>%
  filter(outcome!="Earnings + Allowance") %>%
```

```
  ggplot(data=., aes(x=t, y=att)) +
  geom_point(position = position_dodge(width = 0.4), size = 2.5) +
  geom_linerange(position = position_dodge(width = 0.4),
                 aes(ymin=att-(se*1.96),
                     ymax=att+(se*1.96)),
                 size = 0.7) +
  geom_vline(xintercept = 0, linetype = "longdash") +
  geom_hline(yintercept = 0, linetype = "solid") +
  scale_x_continuous(breaks = seq(-6,3,1), labels = seq(-6,3,1)) +
  scale_y_continuous(breaks = seq(-20000,160000,10000), labels = scales::comma) +
  xlab("Years from first election") +
  ylab("Estimated DiD relative to t-1") +
  theme_bw() +
  coord_cartesian(ylim = c(-20000,80000)) +
  theme(legend.position = c(0.25,0.85),
        legend.title = element_blank(),
        legend.box.background = element_rect(color = "black")) +
  facet_wrap(~outcome)

plot1 / plot2
```

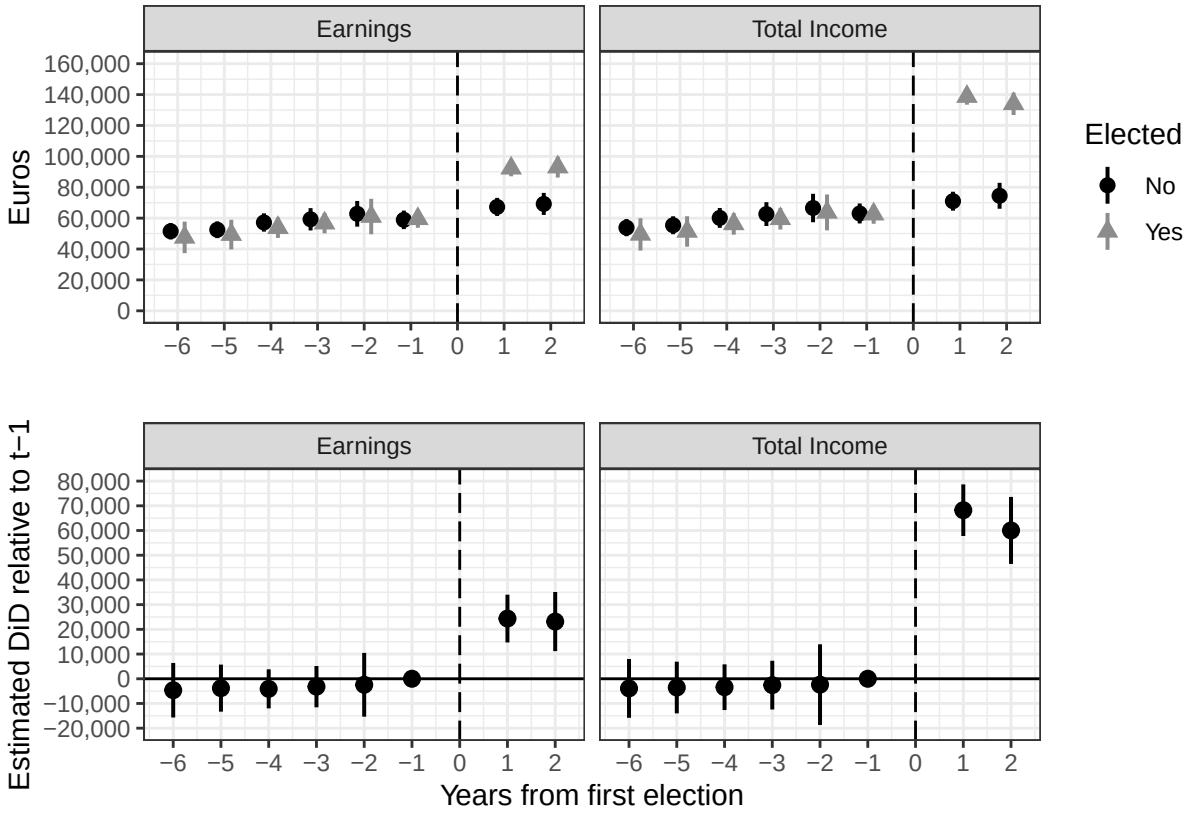


Figure 2

```
load(paste0(work_data, "projects/rewards_or_punishment/data/df_sample_manipulated.rdata"))

df_quantile_sr <- df %>%
  filter(election_year>1993) %>%
  filter(fractile_thres<5) %>%
  filter(t==-2 | t==-1 | t==1) %>%
  as.data.frame()



#### NAIVE OUTCOME
qte_income_p1_est <- panel.qtet(untouched_income~elected,
                                t=1, tmin1 = -1, tmin2 = -2,
                                tname ="t", idname ="PNR",
                                data = df_quantile_sr,
                                probs = seq(0.1,0.9,0.1),
                                se=TRUE, iters=1000)

qte_income_plot_p1 <- ggqte(qte_income_p1_est)

plot_df_naiv <- data.frame(tau=qte_income_p1_est$probs,
                           qte = qte_income_p1_est$qte,
                           qte.lower = qte_income_p1_est$qte.lower,
                           qte.upper = qte_income_p1_est$qte.upper,
                           outcome = "Earnings")

#### FULL OUTCOME
qte_income_pension_p1_est <- panel.qtet(total_income_pension_euro~elected,
                                        t=1, tmin1 = -1, tmin2 = -2,
                                        tname ="t", idname ="PNR",
                                        data = df_quantile_sr,
                                        probs = seq(0.1,0.9,0.1),
                                        se=TRUE, iters=1000)

qte_income_pension_plot_p1 <- ggqte(qte_income_pension_p1_est)

plot_df_pension <- data.frame(tau=qte_income_pension_p1_est$probs,
                              qte = qte_income_pension_p1_est$qte,
                              qte.lower = qte_income_pension_p1_est$qte.lower,
                              qte.upper = qte_income_pension_p1_est$qte.upper,
                              outcome = "Total Income")

plot_qdid_df <- bind_rows(plot_df_naiv, plot_df_pension)

plot_qdid_df %>%
  ggplot(data=., aes(x = tau, y = qte)) +
  geom_point() +
  geom_line() +
  geom_hline(yintercept = 0, linetype = "solid") +
  geom_line(aes(y = qte.upper), linetype = 2) +
  geom_line(aes(y = qte.lower), linetype = 2) +
```

```
scale_x_continuous(breaks = seq(0.1,0.9,0.1), labels = seq(0.1,0.9,0.1)) +
coord_cartesian(ylim = c(-25000,120000)) +
scale_y_continuous(breaks = seq(-30000,120000,10000), label = scales::comma) +
labs(x="Income decile pre-election", y="Winning office's effect on income") +
theme_bw() +
theme(legend.position = c(0.75,0.85), legend.title = element_blank()) +
facet_wrap(~outcome)
```

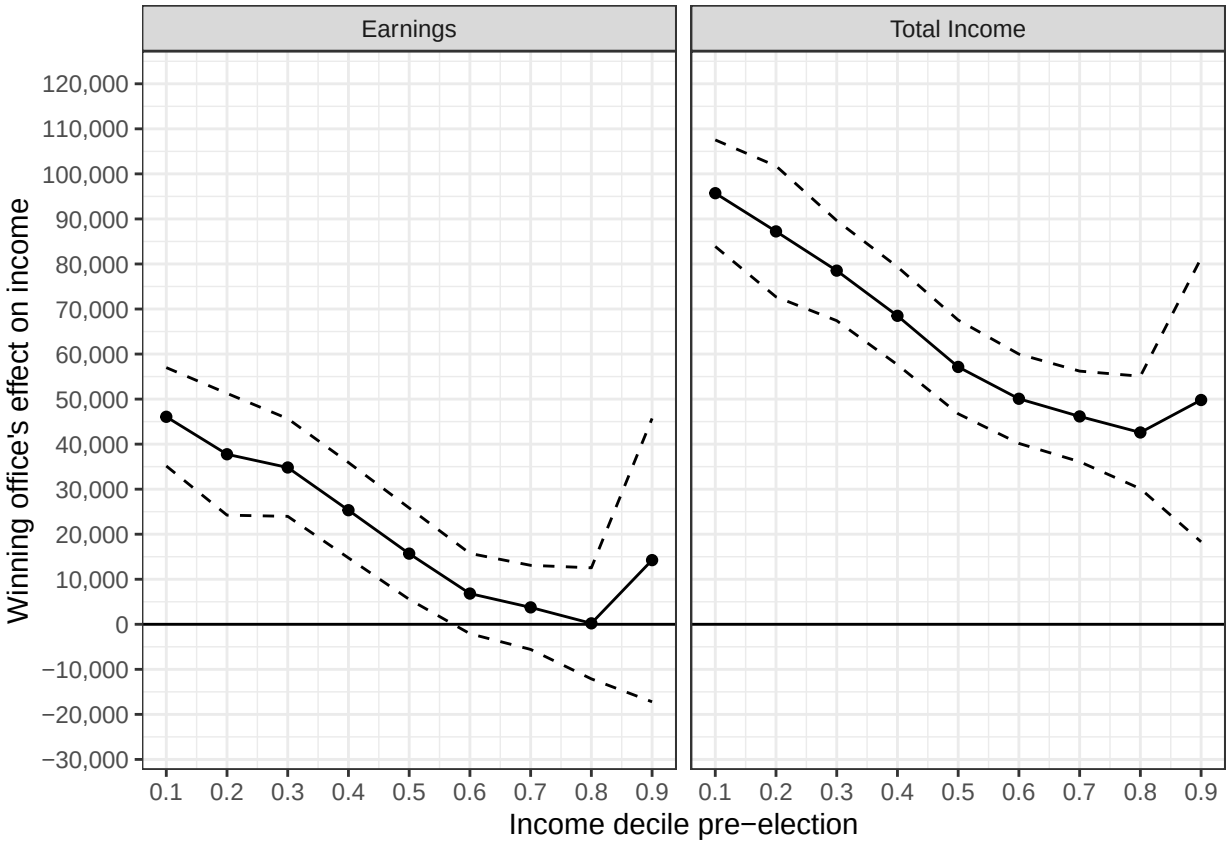
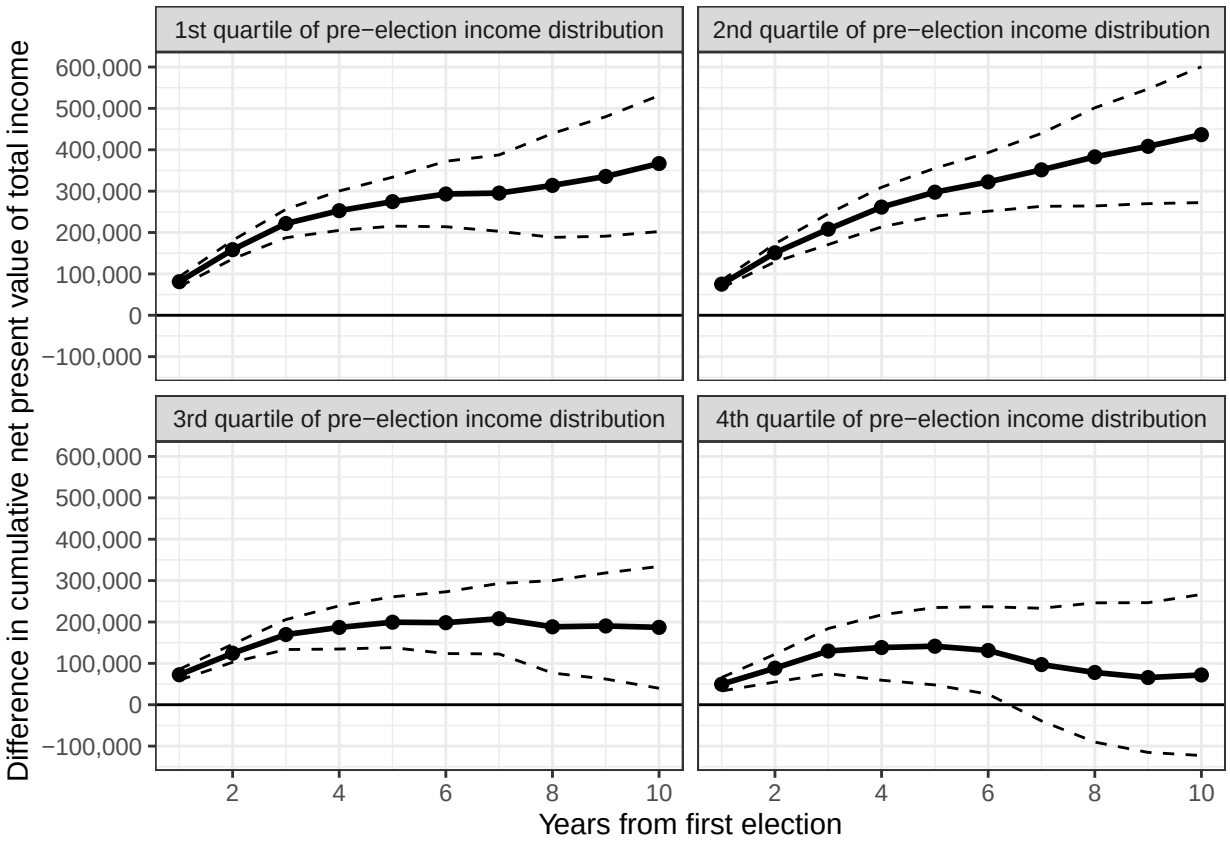


Figure 3

```
df_longrun <- df %>%
  filter(election_year>1993) %>%
  filter(fractile_thres<5) %>%
  filter(t>0)

########################
# create npv variables, for rate of 1,24 and 2,58 and for income inclusive/exclusive pension

### Calculate net present value
df_longrun <- df_longrun %>%
  dplyr::mutate(npv_pens_258 = total_income_pension_euro/(1+0.0258)^t)

# make cumulative net present value
df_longrun <- df_longrun %>%
  group_by(PNR) %>%
  dplyr::mutate(npv_pens_258_cumu = cumsum(npv_pens_258))

### PLOT BY PRE ELECTION INCOME ###
### CALCULATE DIF IN MEANS ACROSS T BY PRE ELECTION INCOME QUARTILE

npv_pens_258_cumu_quartile <- df_longrun %>%
  filter(t>0 & t<21) %>%
  group_by(t, income_pre_percentile) %>%
  do(tidy(lm_robust(npv_pens_258_cumu~elected, data=., clusters = cluster_id))) %>%
  filter(term=="elected")


plot_df_quartile <- npv_pens_258_cumu_quartile %>%
  mutate(income_pre_percentile =
          case_when(income_pre_percentile==1~"1st quartile of pre-election income distribution",
                    income_pre_percentile==2~"2nd quartile of pre-election income distribution",
                    income_pre_percentile==3~"3rd quartile of pre-election income distribution",
                    income_pre_percentile==4~"4th quartile of pre-election income distribution")) %>%
  mutate(income_pre_percentile =
        factor(income_pre_percentile, levels = c("1st quartile of pre-election income distribution",
                                                 "2nd quartile of pre-election income distribution",
                                                 "3rd quartile of pre-election income distribution",
                                                 "4th quartile of pre-election income distribution")))

plot_df_quartile %>%
  filter(t<11) %>%
  ggplot(data=., aes(x = t, y = estimate, fill=outcome, scale = outcome)) +
  geom_point(size = 2, aes(shape = outcome)) +
  geom_line(size=1) +
  geom_line(aes(y = conf.low), linetype = 2) +
  geom_line(aes(y = conf.high), linetype = 2) +
  geom_hline(yintercept = 0, linetype=1) +
  theme_bw() +
  theme(legend.position="none") +
  facet_wrap(~income_pre_percentile, ncol=2) +
  scale_fill_grey(name="", start = 0.75, end = 0.35) +
```

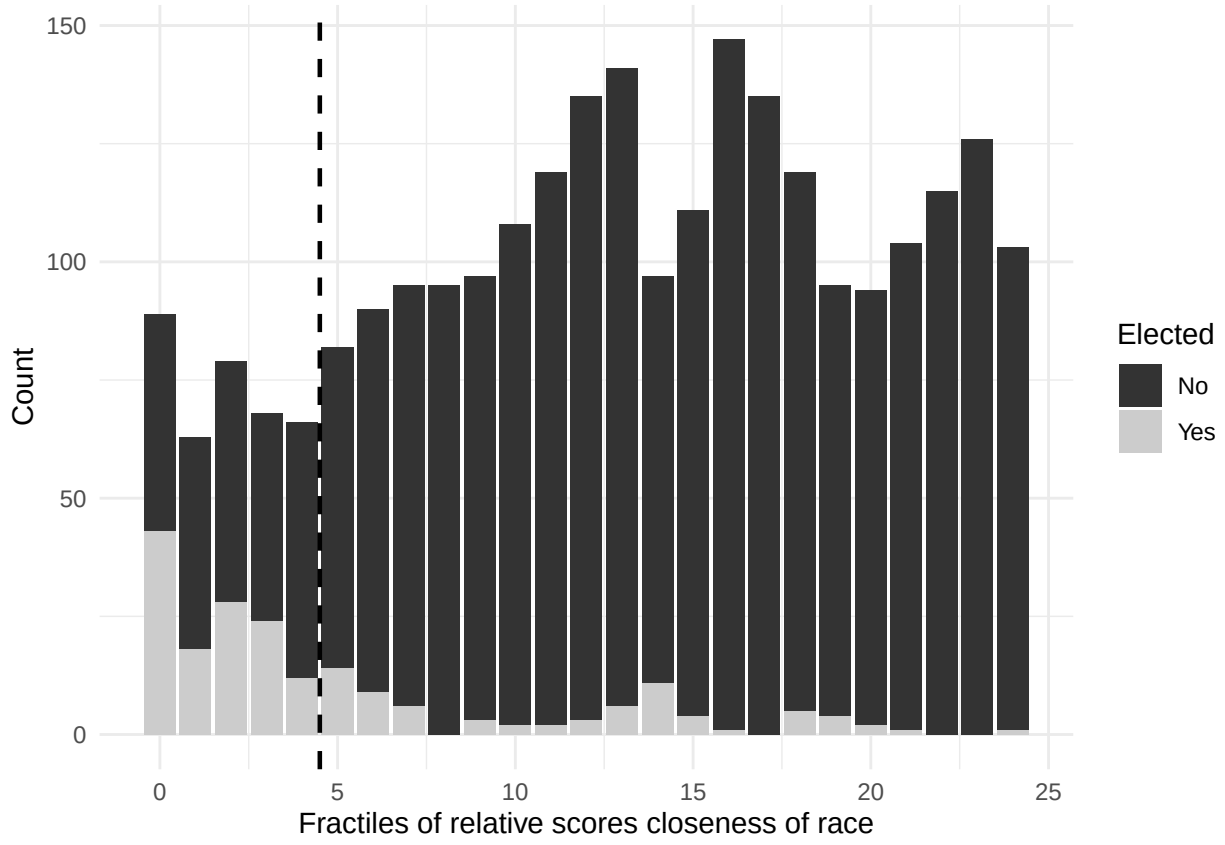
```
  scale_color_grey(name="", start = 0.75, end = 0.35) +
  scale_shape_manual(name="", values =c(19,19,17,17)) +
  guides(alpha="none")  +
  scale_x_continuous(breaks = seq(2,20,2), labels = seq(2,20,2)) +
  coord_cartesian(ylim = c(NA,600000)) +
  scale_y_continuous(breaks = seq(-100000,600000,100000),
                     label = scales::comma) +
  labs(x="Years from first election",
       y="Difference in cumulative net present value of total income")
```



Appendix A

Figure A.1

```
load(paste0(work_data, "projects/rewards_or_punishment/data/candidates_list.rdata"))

all_candidates <- do.call(bind_rows, candidates_list)

### Barplot close to cut-off
all_candidates %>%
  filter(first_run==1 & OPSTILLINGSFORM=="Sideordnet" & election_year>1991 & election_year<2016) %>%
  ggplot(data=., aes(x=fractile_thres, fill = factor_elected)) +
  geom_bar() +
  geom_vline(xintercept = 4.5, linetype = "dashed", size = 0.8) +
  theme_minimal() +
  scale_fill_grey("Elected") +
  xlab("Fractiles of relative scores closeness of race") +
  ylab("Count")
```



Appendix B

Figure B.1

```
load(paste0(work_data, "projects/rewards_or_punishment/data/candidates_list.rdata"))

all_candidates <- do.call(bind_rows, candidates_list)

all_candidates %>%
  filter(first_run==1 & OPSTILLINGSFORM=="Sideordnet" & election_year>1991 & election_year<2016) %>%
  dplyr::mutate(fractile_thres1 = fractile_thres+1) %>%
  dplyr::mutate(running_var = ifelse(elected==0, -fractile_thres1, fractile_thres1)) %>%
  ggplot(data=., aes(x=running_var, fill = factor_elected)) +
  geom_bar() +
  theme_minimal() +
  scale_fill_grey("Elected") +
  xlab("Fractiles of relative scores closeness of race") +
  ylab("Count") +
  geom_vline(xintercept = 0, size = 0.8) +
  geom_vline(xintercept = 5.5, size =0.7, linetype = "dashed") +
  geom_vline(xintercept = -5.5, linetype = "dashed")
```

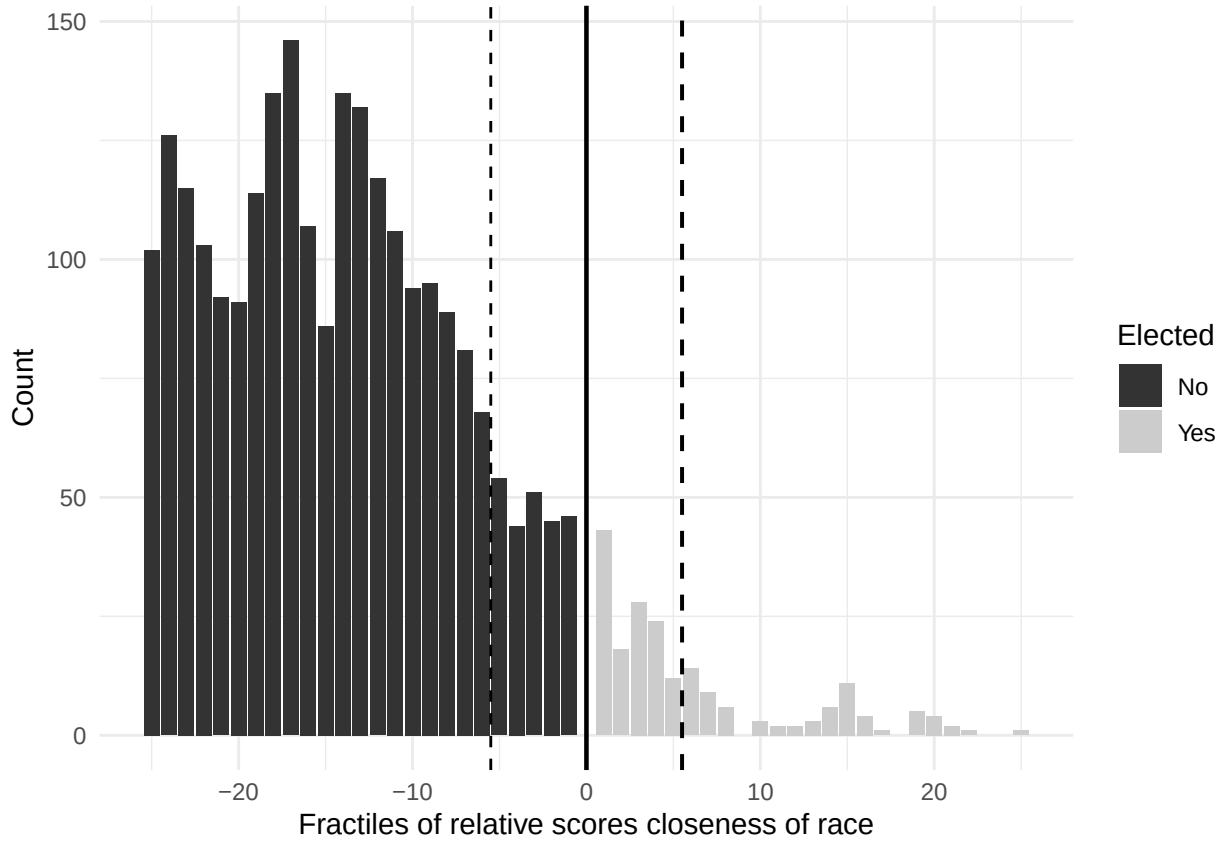


Table B.1

```
# sample of candidates for each year
all_candidates %>%
```

```r
  filter(first_run==1 & fractile_thres<5 & OPSTILLINGSFORM=="Sideordnet" &
           election_year>1991 & election_year < 2016) %>%
  group_by(election_year, elected) %>%
  summarise(n = n())
```

```
## # A tibble: 14 x 3
## # Groups:   election_year [7]
##    election_year elected     n
##            <dbl>   <dbl> <int>
##  1          1994       0    34
##  2          1994       1    16
##  3          1998       0    33
##  4          1998       1    19
##  5          2001       0    47
##  6          2001       1    27
##  7          2005       0    34
##  8          2005       1    20
##  9          2007       0    35
## 10          2007       1    14
## 11          2011       0    25
## 12          2011       1    16
## 13          2015       0    32
## 14          2015       1    13
```

Appendix C

Figure C.1

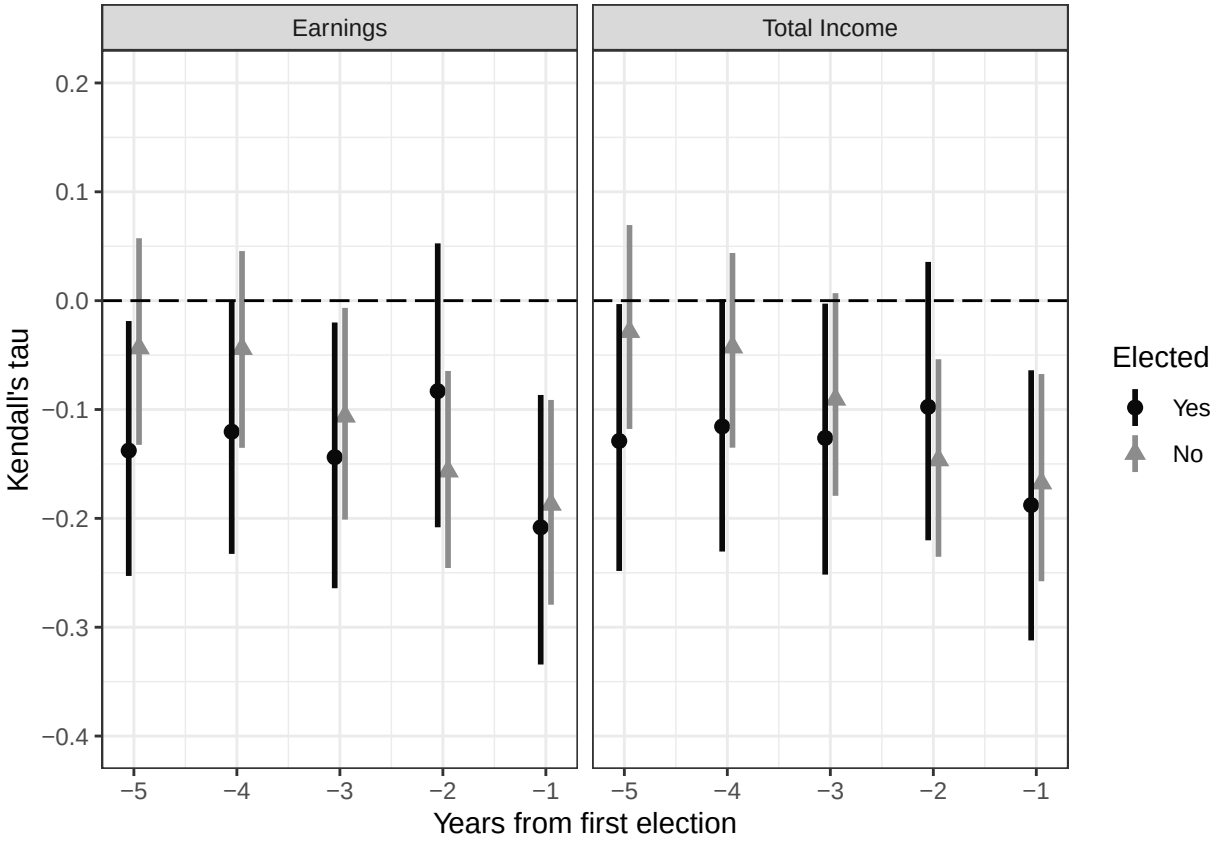
```
load(paste0(work_data, "projects/rewards_or_punishment/data/df_sample_manipulated.rdata"))
df_pt <- df %>%
  filter(election_year>1993) %>%
  filter(fractile_thres<5) %>%
  filter(t!=0) %>%
  filter(t<4) %>%
  as.data.frame()

##### TEST OF COPULA STABILITY ASSUMPTION
df_cop <- df_pt %>%
  group_by(PNR) %>%
  arrange(t) %>%
  dplyr::mutate(total_income_m1 = dplyr::lag(total_income_pension_euro,1)) %>%
  dplyr::mutate(total_income_change = total_income_pension_euro-total_income_m1) %>%
  dplyr::mutate(earnings_m1 = dplyr::lag(untouched_income,1)) %>%
  dplyr::mutate(earnings_change = untouched_income-earnings_m1) %>%
  ungroup() %>%
  filter(t>-6) %>%
  dplyr::mutate(index = t+6)



## bootstrap confidence intervals

######## TOTAL INCOME
## subset of elected
set.seed(3450)
ci_lower <- NA
ci_upper <- NA
kendalls_tau <- NA
index <- NA
for (i in c(1:5, 7:9)){
  df_t <- df_cop %>%
    filter(elected==1) %>%
    filter(!is.na(total_income_m1) & !is.na(total_income_change)) %>%
    filter(index==i) %>%
    as.data.frame()

  x <- as.numeric(df_t$total_income_m1)
  y <- as.numeric(df_t$total_income_change)

  tau <- numeric(0)
  for(b in 1:1000){
    b.sample <- sample(1:length(x), length(x), replace = T)
    options("warn" = -1)
    tau.sample <- cor.test(x[b.sample], y[b.sample], method = "kendall")
    options("warn" = 0)
    tau.sample <- tau.sample$estimate
    tau <- c(tau, tau.sample)
  }
```

```
  tau.hat <- sort(tau)
  ci_lower[i] <- quantile(tau.hat, probs = 0.025)
  ci_upper[i] <- quantile(tau.hat, probs = 0.975)
  index[i] <- i
  kendalls_tau[i] <- cor.test(x, y, method = "kendall")$estimate

}

df_kend_treat_total <- data.frame(index, kendalls_tau, ci_lower, ci_upper)

df_kend_treat_total <- df_kend_treat_total %>%
  filter(!is.na(index)) %>%
  dplyr::mutate(factor_elected = factor("Yes")) %>%
  dplyr::mutate(t = index-6) %>%
  mutate(outcome = "Total Income")



## subset of non-elected
ci_lower <- NA
ci_upper <- NA
kendalls_tau <- NA
index <- NA
for (i in c(1:5, 7:9)){
  df_t <- df_cop %>%
    filter(elected==0) %>%
    filter(!is.na(total_income_m1) & !is.na(total_income_change)) %>%
    filter(index==i) %>%
    as.data.frame()

  x <- as.numeric(df_t$total_income_m1)
  y <- as.numeric(df_t$total_income_change)

  tau <- numeric(0)
  for(b in 1:1000){
    b.sample <- sample(1:length(x), length(x), replace = T)
    options("warn" = -1)
    tau.sample <- cor.test(x[b.sample], y[b.sample], method = "kendall")
    options("warn" = 0)
    tau.sample <- tau.sample$estimate
    tau <- c(tau, tau.sample)
  }
  tau.hat <- sort(tau)
  ci_lower[i] <- quantile(tau.hat, probs = 0.025)
  ci_upper[i] <- quantile(tau.hat, probs = 0.975)
  index[i] <- i
  kendalls_tau[i] <- cor.test(x, y, method = "kendall")$estimate

}

df_kend_control_total <- data.frame(index, kendalls_tau, ci_lower, ci_upper)

df_kend_control_total <- df_kend_control_total %>%
```

```
  filter(!is.na(index)) %>%
  dplyr::mutate(factor_elected = factor("No")) %>%
  dplyr::mutate(t = index-6) %>%
  mutate(outcome = "Total Income")



############################
#### EARNINGS
############

## subset of elected
set.seed(3450)
ci_lower <- NA
ci_upper <- NA
kendalls_tau <- NA
index <- NA
for (i in c(1:5, 7:9)){
  df_t <- df_cop %>%
    filter(elected==1) %>%
    filter(!is.na(earnings_m1) & !is.na(earnings_change)) %>%
    filter(index==i) %>%
    as.data.frame()

  x <- as.numeric(df_t$earnings_m1)
  y <- as.numeric(df_t$earnings_change)

  tau <- numeric(0)
  for(b in 1:1000){
    b.sample <- sample(1:length(x), length(x), replace = T)
    options("warn" = -1)
    tau.sample <- cor.test(x[b.sample], y[b.sample], method = "kendall")
    options("warn" = 0)
    tau.sample <- tau.sample$estimate
    tau <- c(tau, tau.sample)
  }
  tau.hat <- sort(tau)
  ci_lower[i] <- quantile(tau.hat, probs = 0.025)
  ci_upper[i] <- quantile(tau.hat, probs = 0.975)
  index[i] <- i
  kendalls_tau[i] <- cor.test(x, y, method = "kendall")$estimate

}

df_kend_treat_earnings <- data.frame(index, kendalls_tau, ci_lower, ci_upper)

df_kend_treat_earnings <- df_kend_treat_earnings %>%
  filter(!is.na(index)) %>%
  dplyr::mutate(factor_elected = factor("Yes")) %>%
  dplyr::mutate(t = index-6) %>%
  mutate(outcome = "Earnings")
```

```r
## subset of non-elected
ci_lower <- NA
ci_upper <- NA
kendalls_tau <- NA
index <- NA
for (i in c(1:5, 7:9)){
  df_t <- df_cop %>%
    filter(elected==0) %>%
    filter(!is.na(earnings_m1) & !is.na(earnings_change)) %>%
    filter(index==i) %>%
    as.data.frame()

  x <- as.numeric(df_t$earnings_m1)
  y <- as.numeric(df_t$earnings_change)

  tau <- numeric(0)
  for(b in 1:1000){
    b.sample <- sample(1:length(x), length(x), replace = T)
    options("warn" = -1)
    tau.sample <- cor.test(x[b.sample], y[b.sample], method = "kendall")
    options("warn" = 0)
    tau.sample <- tau.sample$estimate
    tau <- c(tau, tau.sample)
  }
  tau.hat <- sort(tau)
  ci_lower[i] <- quantile(tau.hat, probs = 0.025)
  ci_upper[i] <- quantile(tau.hat, probs = 0.975)
  index[i] <- i
  kendalls_tau[i] <- cor.test(x, y, method = "kendall")$estimate

}

df_kend_control_earnings <- data.frame(index, kendalls_tau, ci_lower, ci_upper)

df_kend_control_earnings <- df_kend_control_earnings %>%
  filter(!is.na(index)) %>%
  dplyr::mutate(factor_elected = factor("No")) %>%
  dplyr::mutate(t = index-6) %>%
  mutate(outcome = "Earnings")


# combine the two dfs into one for plotting
df_kend <- bind_rows(df_kend_treat_total, df_kend_control_total,
                     df_kend_treat_earnings, df_kend_control_earnings)

df_kend <- df_kend %>%
  filter(t<0)

# plot of kendalls tau's
df_kend %>%
  ggplot(data = ., aes(x = t, y = kendalls_tau, color = factor_elected)) +
  geom_point(aes(color = factor_elected, shape = factor_elected),
             position = position_dodge(width = 0.2), size = 2.5) +
```

```
  geom_linerange(position = position_dodge(width = 0.2),
                 aes(ymin=ci_lower,
                     ymax=ci_upper),
                 size = 1) +
  geom_hline(yintercept = 0, linetype = "longdash") +
  scale_x_continuous(breaks = seq(-6,-1,1), labels = seq(-6,-1,1)) +
  scale_y_continuous(breaks = seq(-0.5,0.5,0.1), labels = scales::comma) +
  xlab("Years from first election") +
  ylab("Kendall's tau") +
  labs(color="Elected", shape="Elected") +
  scale_color_manual(values = c("grey4", "gray55")) +
  theme_bw() +
  coord_cartesian(ylim = c(-0.4,0.2)) +
  facet_wrap(~outcome)
```



Appendix D

Table D.1

This produces the t-tests that are the content of the table

```
## Define pre-election df
load(paste0(work_data, "projects/rewards_or_punishment/data/df_sample_manipulated.rdata"))


## one year lag
df_1pre <- df %>%
  filter(fractile_thres<5 &
           t==-1 &
           election_year>1993)

t.test(female~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  female by elected
## t = -0.2127, df = 247.25, p-value = 0.8317
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.11859589  0.09547761
## sample estimates:
## mean in group 0 mean in group 1
##       0.3916667       0.4032258
```

```
t.test(age~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  age by elected
## t = -0.54454, df = 209.02, p-value = 0.5866
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -3.447186  1.954981
## sample estimates:
## mean in group 0 mean in group 1
##        39.24583        39.99194
```

```
t.test(age2~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  age2 by elected
## t = -1.0413, df = 204.46, p-value = 0.299
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -337.7420  104.2922
## sample estimates:
## mean in group 0 mean in group 1
##        1654.654        1771.379
```

```r
t.test(married~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  married by elected
## t = 0.4717, df = 247.29, p-value = 0.6376
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.08280386  0.13495439
## sample estimates:
## mean in group 0 mean in group 1
##       0.5583333       0.5322581
```

```r
t.test(untouched_income~elected,df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  untouched_income by elected
## t = -0.15714, df = 331.29, p-value = 0.8752
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -9046.460  7708.046
## sample estimates:
## mean in group 0 mean in group 1
##        58894.33        59563.54
```

```r
t.test(total_income_pension_euro~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  total_income_pension_euro by elected
## t = 0.10465, df = 333.2, p-value = 0.9167
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -8507.709  9463.759
## sample estimates:
## mean in group 0 mean in group 1
##        63028.06        62550.03
```

```r
t.test(avg_income_pension_pre~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  avg_income_pension_pre by elected
## t = 0.56359, df = 287.8, p-value = 0.5735
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -6693.521 12064.881
## sample estimates:
## mean in group 0 mean in group 1
##        60068.62        57382.94
```

```r
t.test(mother_percentile~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  mother_percentile by elected
## t = -1.6838, df = 59.451, p-value = 0.09746
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -19.791091   1.701865
## sample estimates:
## mean in group 0 mean in group 1
##        58.04630        67.09091
```

```r
t.test(rightwing~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  rightwing by elected
## t = 0.65421, df = 245.43, p-value = 0.5136
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.07188996  0.14339533
## sample estimates:
## mean in group 0 mean in group 1
##       0.6083333       0.5725806
```

```r
t.test(majority_coalition~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  majority_coalition by elected
## t = -0.43638, df = 248.59, p-value = 0.6629
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.13338899  0.08500189
## sample estimates:
## mean in group 0 mean in group 1
##       0.5000000       0.5241935
```

```r
t.test(primaryschooleducation~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  primaryschooleducation by elected
## t = 0.11089, df = 251.43, p-value = 0.9118
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.06757929  0.07564381
## sample estimates:
## mean in group 0 mean in group 1
##       0.1250000       0.1209677
```

```r
t.test(highschooleducation~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  highschooleducation by elected
## t = 1.207, df = 272.6, p-value = 0.2285
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.03231505  0.13473440
## sample estimates:
## mean in group 0 mean in group 1
##       0.2125000       0.1612903
```

```r
t.test(vocationaleducation~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  vocationaleducation by elected
## t = 0.046676, df = 249.24, p-value = 0.9628
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.08305540  0.08708766
## sample estimates:
## mean in group 0 mean in group 1
##       0.1875000       0.1854839
```

```r
t.test(mediumlongeducation~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  mediumlongeducation by elected
## t = -0.95671, df = 237.31, p-value = 0.3397
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.14720064  0.05096408
## sample estimates:
## mean in group 0 mean in group 1
##       0.2583333       0.3064516
```

```r
t.test(longeducation~elected, df_1pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  longeducation by elected
## t = -0.19797, df = 245.13, p-value = 0.8432
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.10007498  0.08179541
## sample estimates:
## mean in group 0 mean in group 1
##       0.2166667       0.2258065
```

Figure D.1

```
load(paste0(work_data, "projects/rewards_or_punishment/data/df_sample_manipulated.rdata"))

df_pre <- df %>%
  filter(fractile_thres<5 &
           t<0 &
           election_year>1993 &
           election_year<2019) %>%
  group_by(PNR, factor_elected, party_factor, election_year_factor) %>%
  summarise(mean_pre_income_total = mean(total_income_pension_euro, na.rm=T)/1000,
            avg_earnings = mean(untouched_income, na.rm = TRUE)/1000)

p1 <- df_pre %>%
  ggplot(data=., aes(x = mean_pre_income_total, fill = factor_elected)) +
  geom_density(alpha=0.5) +
  coord_cartesian(xlim = c(-70,200)) +
  scale_x_continuous(breaks = seq(-50,175,25), label = scales::comma) +
  xlab("Total income average over t-6 to t-1") +
  ylab("Density") +
  scale_fill_grey("Elected") +
  ggtitle("Total Income") +
  theme_bw() +
  theme(axis.text.x = element_text(angle=90,vjust=0.5,hjust = 1)) +
  facet_wrap(~election_year_factor, ncol=2)

p2 <- df_pre %>%
  ggplot(data=., aes(x = avg_earnings, fill = factor_elected)) +
  geom_density(alpha=0.5) +
  coord_cartesian(xlim = c(-70,200)) +
  scale_x_continuous(breaks = seq(-50,175,25), label = scales::comma) +
  xlab("Earnings average over t-6 to t-1") +
  ylab("Density") +
  scale_fill_grey("Elected") +
  ggtitle("Earnings") +
  theme_bw() +
  theme(axis.text.x = element_text(angle=90,vjust=0.5,hjust = 1), legend.position = "none") +
  facet_wrap(~election_year_factor, ncol=2)

p2+p1
```

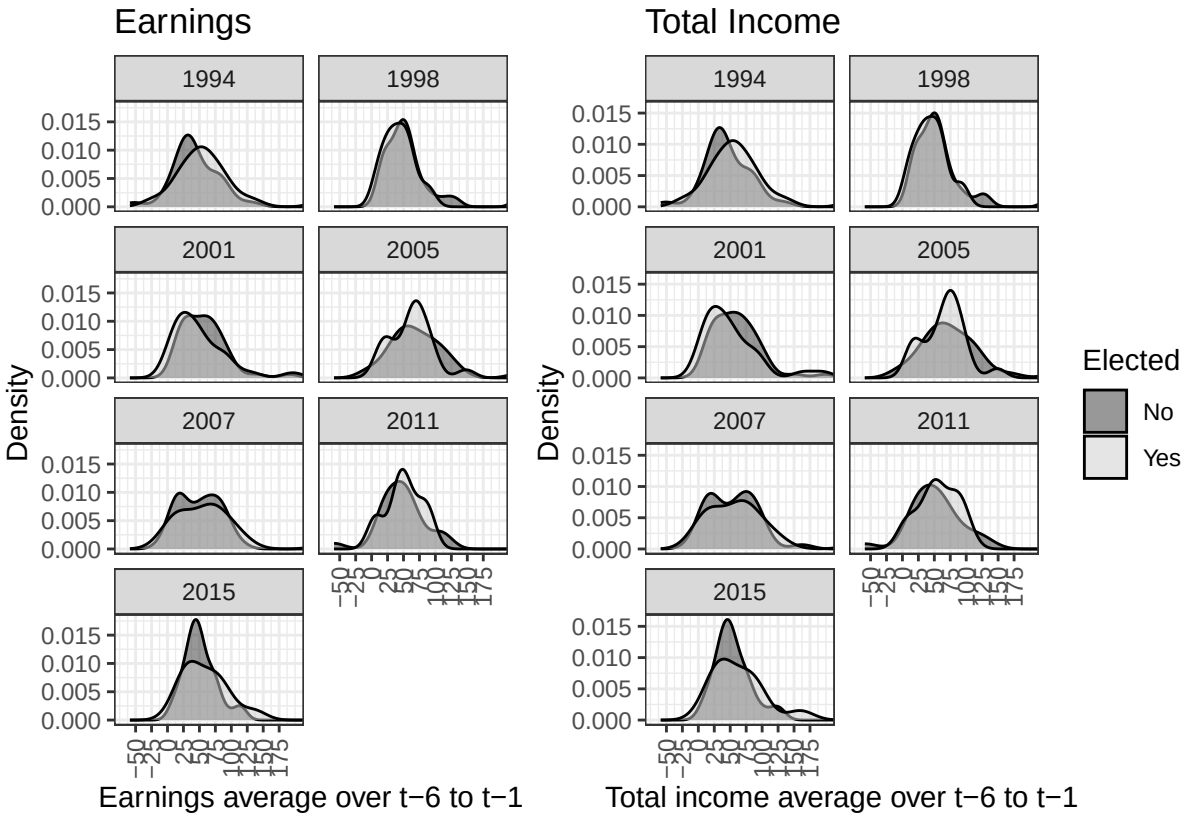


Figure D.2

```
load(paste0(work_data, "projects/rewards_or_punishment/data/df_sample_manipulated.rdata"))

df_pt <- df %>%
  filter(election_year>1993) %>%
  filter(fractile_thres<5) %>%
  filter(t!=0) %>%
  filter(t<4) %>%
  as.data.frame()

# and remove years that are post election (t)
df_pt_pre <- df_pt %>%
  filter(t<0)


# cant find a loop solution - which is quite annoying
# filter() is not feasible because it's two conditions and filter() cannot
# take strings as an input -- so the solution below is a long but does the job


###### TOTAL INCOME
placebo_list <- list()

placebo_list[[1]] <- df_pt_pre %>%
```

```
  filter(t==-6 | t==-5) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-6 vs -5")

placebo_list[[2]] <- df_pt_pre %>%
  filter(t==-6 | t==-4) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-6 vs -4")

placebo_list[[3]] <- df_pt_pre %>%
  filter(t==-6 | t==-3) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-6 vs -3")

placebo_list[[4]] <- df_pt_pre %>%
  filter(t==-6 | t==-2) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-6 vs -2")

placebo_list[[5]] <- df_pt_pre %>%
  filter(t==-6 | t==-1) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-6 vs -1")

placebo_list[[6]] <- df_pt_pre %>%
  filter(t==-5 | t==-4) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-5 vs -4")

placebo_list[[7]] <- df_pt_pre %>%
  filter(t==-5 | t==-3) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-5 vs -3")

placebo_list[[8]] <- df_pt_pre %>%
  filter(t==-5 | t==-2) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-5 vs -2")

placebo_list[[9]] <- df_pt_pre %>%
  filter(t==-5 | t==-1) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-5 vs -1")
```

```r
placebo_list[[10]] <- df_pt_pre %>%
  filter(t==-4 | t==-3) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-4 vs -3")

placebo_list[[11]] <- df_pt_pre %>%
  filter(t==-4 | t==-2) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-4 vs -2")

placebo_list[[12]] <- df_pt_pre %>%
  filter(t==-4 | t==-1) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-4 vs -1")

placebo_list[[13]] <- df_pt_pre %>%
  filter(t==-3 | t==-2) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-3 vs -2")

placebo_list[[14]] <- df_pt_pre %>%
  filter(t==-3 | t==-1) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-3 vs -1")

placebo_list[[15]] <- df_pt_pre %>%
  filter(t==-2 | t==-1) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(total_income_pension_euro~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-2 vs -1")

placebo_df_total <- do.call(bind_rows, placebo_list)

placebo_df_total <- placebo_df_total %>%
  filter(term=="elected:placebo_post") %>%
  mutate(outcome = "Total Income")

###### earnings
placebo_list_earnings <- list()

placebo_list_earnings[[1]] <- df_pt_pre %>%
  filter(t==-6 | t==-5) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-6 vs -5")

placebo_list_earnings[[2]] <- df_pt_pre %>%
  filter(t==-6 | t==-4) %>%
```

```
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-6 vs -4")

placebo_list_earnings[[3]] <- df_pt_pre %>%
  filter(t==-6 | t==-3) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-6 vs -3")

placebo_list_earnings[[4]] <- df_pt_pre %>%
  filter(t==-6 | t==-2) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-6 vs -2")

placebo_list_earnings[[5]] <- df_pt_pre %>%
  filter(t==-6 | t==-1) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-6 vs -1")

placebo_list_earnings[[6]] <- df_pt_pre %>%
  filter(t==-5 | t==-4) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-5 vs -4")

placebo_list_earnings[[7]] <- df_pt_pre %>%
  filter(t==-5 | t==-3) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-5 vs -3")

placebo_list_earnings[[8]] <- df_pt_pre %>%
  filter(t==-5 | t==-2) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-5 vs -2")

placebo_list_earnings[[9]] <- df_pt_pre %>%
  filter(t==-5 | t==-1) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-5 vs -1")

placebo_list_earnings[[10]] <- df_pt_pre %>%
  filter(t==-4 | t==-3) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-4 vs -3")

placebo_list_earnings[[11]] <- df_pt_pre %>%
```

```
  filter(t==-4 | t==-2) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-4 vs -2")

placebo_list_earnings[[12]] <- df_pt_pre %>%
  filter(t==-4 | t==-1) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-4 vs -1")

placebo_list_earnings[[13]] <- df_pt_pre %>%
  filter(t==-3 | t==-2) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-3 vs -2")

placebo_list_earnings[[14]] <- df_pt_pre %>%
  filter(t==-3 | t==-1) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-3 vs -1")

placebo_list_earnings[[15]] <- df_pt_pre %>%
  filter(t==-2 | t==-1) %>%
  dplyr::mutate(placebo_post = ifelse(t==max(t),1,0)) %>%
  do(tidy(lm(untouched_income~elected*placebo_post, data=.))) %>%
  dplyr::mutate(comparison = "-2 vs -1")

placebo_df_earnings <- do.call(bind_rows, placebo_list_earnings)

placebo_df_earnings <- placebo_df_earnings %>%
  filter(term=="elected:placebo_post") %>%
  mutate(outcome = "Earnings")

## merge
placebo_df <- bind_rows(placebo_df_total, placebo_df_earnings)

## plot coefs
placebo_df %>%
  ggplot(data =., aes(x = comparison, y = estimate)) +
  geom_point(col = "black",
             size = 2) +
  geom_hline(yintercept = 0, linetype="dashed") +
  geom_linerange(aes(ymin = estimate-(std.error*1.96),
                     ymax = estimate+(std.error*1.96)),
                 col = "black",
                 size = 1.2) +
  theme_bw() +
  xlab("Placebo DiD estimates between winners and losers") +
  ylab("Pre-election year comparison") +
  theme(axis.text.x = element_text(angle=45,hjust = 1)) +
  coord_flip(ylim = c(-40000,40000)) +
```

```
  scale_x_discrete(limits = rev(levels(placebo_df$comparison))) +
  scale_y_continuous(breaks = seq(-40000,40000,5000), label = scales::comma) +
  facet_wrap(~outcome)
```

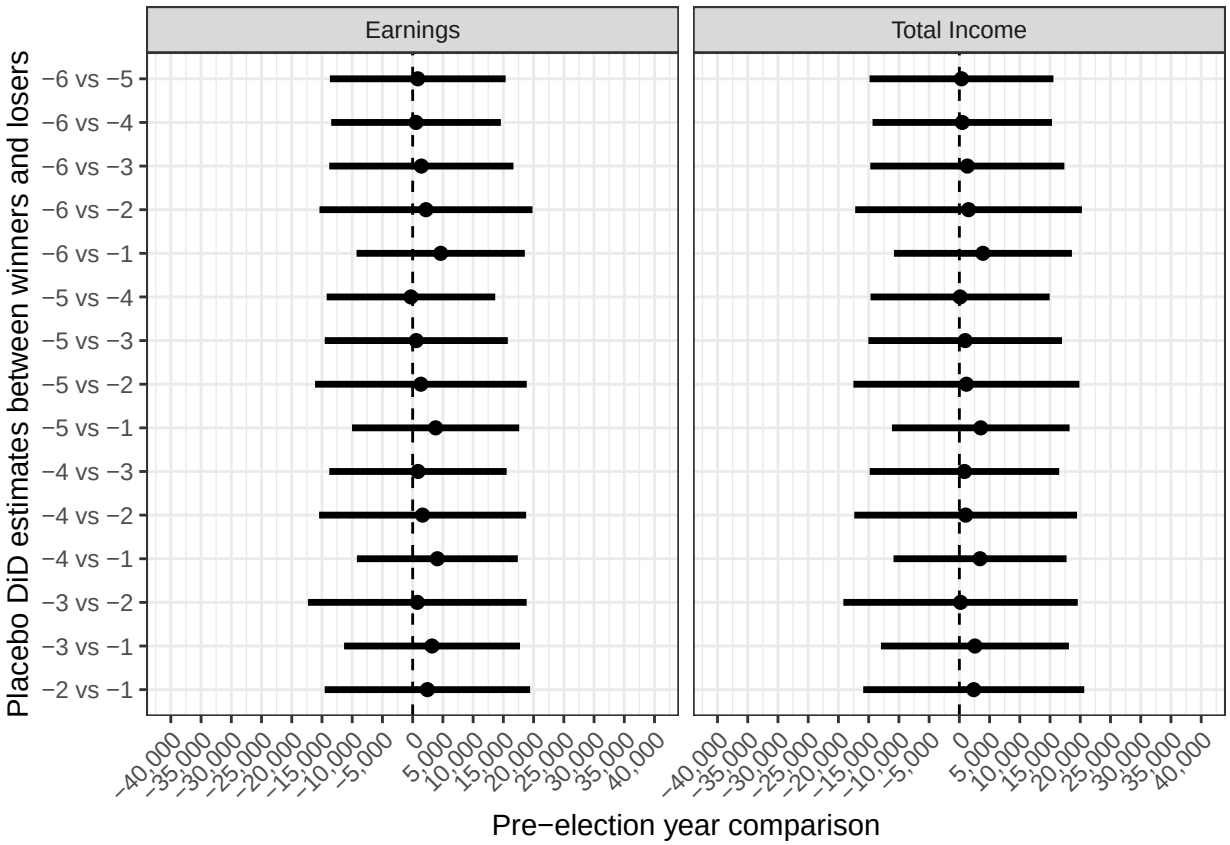


Figure D.3

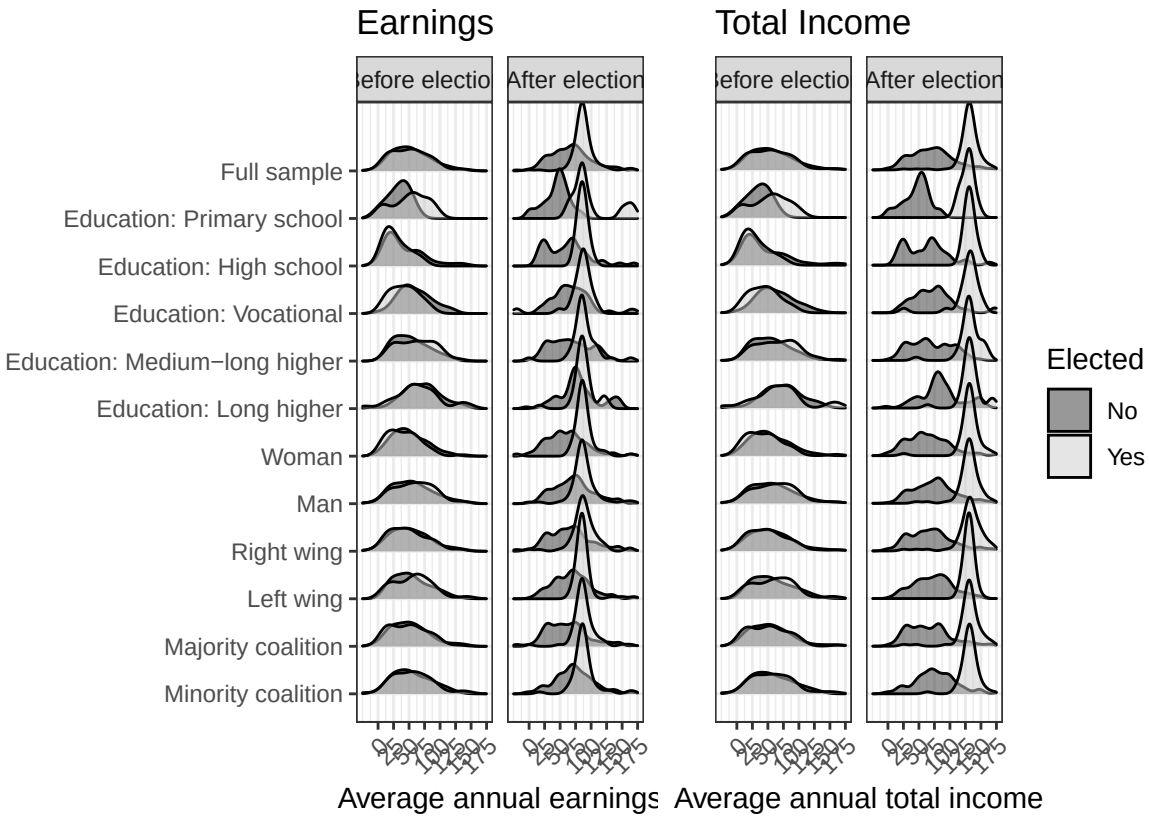
```
load(paste0(work_data, "projects/rewards_or_punishment/data/df_sample_manipulated.rdata"))

df_longrun <- df %>%
  filter(election_year>1993) %>%
  filter(fractile_thres<5) %>%
  filter(t!=0) %>%
  filter(t>-6) %>%
  as.data.frame()



# calculate post-
plot_pre_dens_total <- df_longrun %>%
  filter(t==-1) %>%
  mutate(avg_income = avg_income_pension_pre/1000) %>%
  dplyr::select(c(PNR, avg_income, factor_elected, female, rightwing, majority_coalition,
                  post_election, primaryschooleducation, vocationaleducation,
                  highschooleducation, mediumlongeducation, longeducation))
```

```r
plot_post_dens_total <- df_longrun %>%
  filter(post_election==1) %>%
  group_by(PNR, factor_elected, female, rightwing, majority_coalition, post_election,
           primaryschooleducation, vocationaleducation, highschooleducation, mediumlongeducation,
           longeducation) %>%
  summarise(avg_income = mean(total_income_pension_euro, na.rm = TRUE)/1000)

# bind pre and post dfs
plot_dens_total <- bind_rows(plot_pre_dens_total, plot_post_dens_total)

# make dfs long and tidy
plot_ridge_total <-
  pivot_longer(plot_dens_total,
               cols = c(female, rightwing, majority_coalition, primaryschooleducation,
                        vocationaleducation, highschooleducation, mediumlongeducation,
                        longeducation), names_to = "covariate")

# aadd this to make an overall distribution as well
plot_overall_total <- plot_dens_total %>%
  dplyr::select(PNR, factor_elected, post_election, avg_income) %>%
  dplyr::mutate(covariate = "overall",
                value = 1)


plot_ridge_total <- bind_rows(plot_overall_total, plot_ridge_total) %>%
  mutate(outcome = "Total Income")

plot_ridge_total <- plot_ridge_total %>%
  mutate(labels = case_when(covariate=="overall" & value==1~"Full sample",
                            covariate=="female" & value==1~"Woman",
                            covariate=="female" & value==0~"Man",
                            covariate=="rightwing" & value==1~"Right wing",
                            covariate=="rightwing" & value==0~"Left wing",
                            covariate=="majority_coalition" & value==1~"Majority coalition",
                            covariate=="majority_coalition" & value==0~"Minority coalition",
                            covariate=="primaryschooleducation" & value==1~"Education: Primary school",
                            covariate=="highschooleducation" & value==1~"Education: High school",
                            covariate=="vocationaleducation" & value==1~"Education: Vocational",
                            covariate=="mediumlongeducation" & value==1~"Education: Medium-long higher"
                            covariate=="longeducation" & value==1~"Education: Long higher")) %>%
  ## next up decides the order in the plot - counterintuitively, first levels are placed in the bottom
  dplyr::mutate(labels = factor(labels, levels = c("Minority coalition",
                                                   "Majority coalition",
                                                   "Left wing",
                                                   "Right wing",
                                                   "Man",
                                                   "Woman",
                                                   "Education: Long higher",
                                                   "Education: Medium-long higher",
                                                   "Education: Vocational",
                                                   "Education: High school",
                                                   "Education: Primary school",
                                                   "Full sample"
```

```
  )))

# remove NA's created because of the education dummies in case_when (with no TRUE~0)
plot_ridge_total <- plot_ridge_total %>%
  filter(!is.na(labels))




#############EARNINGS

# calculate post-
plot_pre_dens_earnings <- df_longrun %>%
  filter(t==-1) %>%
  mutate(avg_income = avg_earnings_pre/1000) %>%
  dplyr::select(c(PNR, avg_income, factor_elected, female, rightwing, majority_coalition,
                  post_election, primaryschooleducation, vocationaleducation,
                  highschooleducation, mediumlongeducation, longeducation))


plot_post_dens_earnings <- df_longrun %>%
  filter(post_election==1) %>%
  group_by(PNR, factor_elected, female, rightwing, majority_coalition, post_election,
           primaryschooleducation, vocationaleducation, highschooleducation, mediumlongeducation,
           longeducation) %>%
  summarise(avg_income = mean(untouched_income, na.rm = TRUE)/1000)

# bind pre and post dfs
plot_dens_earnings <- bind_rows(plot_pre_dens_earnings, plot_post_dens_earnings)

# make dfs long and tidy
plot_ridge_earnings <- pivot_longer(plot_dens_earnings,
                                    cols = c(female, rightwing, majority_coalition, primaryschooleducati
                                             vocationaleducation, highschooleducation, mediumlongeducati
                                             longeducation), names_to = "covariate")

# aadd this to make an overall distribution as well
plot_overall_earnings <- plot_dens_earnings %>%
  dplyr::select(PNR, factor_elected, post_election, avg_income) %>%
  dplyr::mutate(covariate = "overall",
                value = 1)


plot_ridge_earnings <- bind_rows(plot_overall_earnings, plot_ridge_earnings) %>%
  mutate(outcome = "Earnings")

plot_ridge_earnings <- plot_ridge_earnings %>%
  mutate(labels = case_when(covariate=="overall" & value==1~"Full sample",
                            covariate=="female" & value==1~"Woman",
                            covariate=="female" & value==0~"Man",
                            covariate=="rightwing" & value==1~"Right wing",
                            covariate=="rightwing" & value==0~"Left wing",
                            covariate=="majority_coalition" & value==1~"Majority coalition",
```

```
                              covariate=="majority_coalition" & value==0~"Minority coalition",
                              covariate=="primaryschooleducation" & value==1~"Education: Primary school",
                              covariate=="highschooleducation" & value==1~"Education: High school",
                              covariate=="vocationaleducation" & value==1~"Education: Vocational",
                              covariate=="mediumlongeducation" & value==1~"Education: Medium-long higher"
                              covariate=="longeducation" & value==1~"Education: Long higher")) %>%
  ## next up decides the order in the plot - counterintuitively, first levels are placed in the bottom
  dplyr::mutate(labels = factor(labels, levels = c("Minority coalition",
                                                   "Majority coalition",
                                                   "Left wing",
                                                   "Right wing",
                                                   "Man",
                                                   "Woman",
                                                   "Education: Long higher",
                                                   "Education: Medium-long higher",
                                                   "Education: Vocational",
                                                   "Education: High school",
                                                   "Education: Primary school",
                                                   "Full sample"
  )))

# remove NA's created because of the education dummies in case_when (with no TRUE~0)
plot_ridge_earnings <- plot_ridge_earnings %>%
  filter(!is.na(labels))


## create the two plots seperately and then combine at last



p1 <-
plot_ridge_total %>%
  ggplot(data=., aes(x = avg_income, y = labels, fill = factor_elected)) +
  geom_density_ridges(alpha = 0.5, panel_scaling = FALSE) +
  facet_grid(~post_election,
             labeller = as_labeller(c(`1`="After election", `0`="Before election")), scales = "fixed") +
  scale_x_continuous(breaks = seq(0,175,25), limits = c(-25,175), label = scales::comma) +
  xlab("Average annual total income") +
  ylab("") +
  scale_fill_grey("Elected") +
  ggtitle("Total Income") +
  theme_bw() +
  theme(axis.text.x = element_text(angle=45,hjust = 1), axis.text.y = element_blank(),
        axis.ticks.y = element_blank())

p2 <-
  plot_ridge_earnings %>%
  ggplot(data=., aes(x = avg_income, y = labels, fill = factor_elected)) +
  geom_density_ridges(alpha = 0.5, panel_scaling = FALSE) +
  facet_grid(~post_election,
             labeller = as_labeller(c(`1`="After election", `0`="Before election")), scales = "fixed") +
  scale_x_continuous(breaks = seq(0,175,25), limits = c(-25,175), label = scales::comma) +
  xlab("Average annual earnings") +
```

```
  ylab("") +
  scale_fill_grey("Elected") +
  ggtitle("Earnings") +
  theme_bw() +
  theme(axis.text.x = element_text(angle=45,hjust = 1),legend.position = "none")

p2+p1
```



Appendix E

Table E.1

```
load(paste0(work_data, "projects/rewards_or_punishment/data/df_sample_manipulated.rdata"))

df_shortrun <- df %>%
  filter(election_year>1993) %>%
  filter(fractile_thres<5) %>%
  filter(t>=-6) %>%
  filter(t!=0) %>%
  filter(t<3) %>%
  as.data.frame()

############################
#### TOTAL INCOME ####
############################
sr_income_pension <-
  felm(total_income_pension_euro~did + post_election + elected |0|0|cluster_id, data = df_shortrun)

sr_income_pension_fe <-
  felm(total_income_pension_euro~did + post_election + elected + cluster_id |0|0|cluster_id,
       data = df_shortrun)

stargazer(sr_income_pension, sr_income_pension_fe,
          type = "text",
          digits = 0,
          keep = c("did", "post_election", "elected", "Constant"),
          covariate.labels = c("textbf{Elected times Post Election (DiD)}",
                               "Post Election", "Elected"),
          dep.var.caption = "Total Income + Allowance + Pension (2015 euro)",
          dep.var.labels.include = FALSE,
          keep.stat = c("n"),
          add.lines = list(c("Fixed effects for cluster?", "NO", "YES")),
          title = "Short Run Average Returns to Office - including allowance and pension")
```

```
##
## Short Run Average Returns to Office - including allowance and pension
## ================================================================================================
##                                          Total Income + Allowance + Pension (2015 euro)
##                                          ----------------------------------------------
##                                                (1)                      (2)
## ------------------------------------------------------------------------------------------------
## textbf{Elected times Post Election (DiD)}       70,273***               69,410***
##                                                  (4,199)                  (4,114)
##
## Post Election                                    10,470***               12,850***
##                                                  (2,773)                  (2,623)
##
## Elected                                          -2,536                   1,615
##                                                  (4,853)                  (6,641)
##
## Constant                                         60,953***               25,334***
##                                                  (3,240)                  (6,315)
```

```
##
## ---------------------------------------------------------------------------------------
## Fixed effects for cluster?                          NO                      YES
## Observations                                      2,896                   2,896
## ===================================================================================
## Note:                                                        *p<0.1; **p<0.05; ***p<0.01
```

Table E.2

```r
### EARNINGS

sr_income <- felm(untouched_income~did + post_election + elected |0|0|cluster_id, data = df_shortrun)
sr_income_fe <- felm(untouched_income~did + post_election + elected + cluster_id |0|0|cluster_id,
                     data = df_shortrun)

stargazer(sr_income, sr_income_fe,
          type = "text",
          digits = 0,
          keep = c("did", "post_election", "elected", "Constant"),
          covariate.labels = c("textbf{Elected times Post Election (DiD)}",
                               "Post Election", "Elected"),
          dep.var.caption = "Total Income (2015 euro)",
          dep.var.labels.include = FALSE,
          keep.stat = c("n"),
          add.lines = list(c("Fixed effects for cluster?", "NO", "YES")),
          title = "Short Run Average Returns to Office - excluding allowance and pension")
```

```
##
## Short Run Average Returns to Office - excluding allowance and pension
## ====================================================================
##                                             Total Income (2015 euro)
##                                           ----------------------------
##                                                (1)            (2)
## --------------------------------------------------------------------
## textbf{Elected times Post Election (DiD)}   26,461***      25,732***
##                                              (3,962)        (3,920)
##
## Post Election                               10,336***      12,408***
##                                              (2,592)        (2,522)
##
## Elected                                       -1,698         1,267
##                                              (4,524)        (6,325)
##
## Constant                                    57,425***      25,284***
##                                              (2,936)        (6,040)
##
## --------------------------------------------------------------------
## Fixed effects for cluster?                      NO            YES
## Observations                                  2,896          2,896
## ====================================================================
## Note:                                      *p<0.1; **p<0.05; ***p<0.01
```

Table E.3

```
##############################
###### DISPOSABLE EARNINGS ####
##############################

sr_disp_income <-
  felm(income_m_taxes_euro~did + post_election + elected |0|0|cluster_id, data = df_shortrun)

sr_disp_income_fe <-
  felm(income_m_taxes_euro~did + post_election + elected + cluster_id |0|0|cluster_id,
       data = df_shortrun)

stargazer(sr_disp_income, sr_disp_income_fe,
          type = "text",
          digits = 0,
          keep = c("did", "post_election", "elected", "Constant"),
          covariate.labels = c("textbf{Elected times Post Election (DiD)}",
                               "Post Election", "Elected"),
          dep.var.caption = "Disposable Income (2015 euro)",
          dep.var.labels.include = FALSE,
          keep.stat = c("n"),
          add.lines = list(c("Fixed effects for cluster?", "NO", "YES")),
          title = "Short Run Average Returns to Office - Disposable Income")
```

```
##
## Short Run Average Returns to Office - Disposable Income
## ================================================================================
##                                            Disposable Income (2015 euro)
##                                            -------------------------------
##                                                  (1)             (2)
## --------------------------------------------------------------------------------
## textbf{Elected times Post Election (DiD)}    21,492***       21,257***
##                                               (2,819)         (2,950)
##
## Post Election                                6,736***        7,821***
##                                               (1,555)         (1,594)
##
## Elected                                       -3,495          -1,201
##                                               (2,828)         (3,480)
##
## Constant                                     32,746***       18,057***
##                                               (1,519)         (3,288)
##
## --------------------------------------------------------------------------------
## Fixed effects for cluster?                      NO              YES
## Observations                                  2,896           2,896
## ================================================================================
## Note:                                        *p<0.1; **p<0.05; ***p<0.01
```

Table E.4

```
sr1_perks <-
  felm(perks_euro~did + post_election + elected |0|0|cluster_id, data = df_shortrun)
```

```r
sr2_perks <-
  felm(perks_euro~did + post_election + elected + cluster_id |0|0|cluster_id,
       data = df_shortrun)

sr1_taxfreeincome <-
  felm(taxfree_income_euro~did + post_election + elected |0|0|cluster_id, data = df_shortrun)

sr2_taxfreeincome <-
  felm(taxfree_income_euro~did + post_election + elected + cluster_id |0|0|cluster_id,
       data = df_shortrun)

stargazer(sr1_perks, sr2_perks, sr1_taxfreeincome, sr2_taxfreeincome,
          type = "text",
          digits = 0,
          keep = c("did", "post_election", "elected", "Constant"),
          covariate.labels = c("textbf{Elected times Post Election (DiD)}",
                               "Post Election", "Elected"),
          dep.var.labels = c("Perks (2015 euro)", "Taxfree Income (2015 euro)"),
          keep.stat = c("n"),
          add.lines = list(c("Fixed effects for cluster?", "NO", "YES", "NO", "YES")),
          title = "Tax-excempted allowances not covered by administrative measurements of perks and tax
```

```
##
## Tax-excempted allowances not covered by administrative measurements of perks and taxfree income
## ==================================================================================
##                                                      Dependent variable:
##                                          ---------------------------------------------
##                                            Perks (2015 euro)     Taxfree Income (2015 euro)
##                                            (1)         (2)           (3)        (4)
## ----------------------------------------------------------------------------------
## textbf{Elected times Post Election (DiD)}   -52         -151          138        89
##                                           (246)       (239)          (97)      (101)
##
## Post Election                               -8          141         -216***    -204***
##                                           (188)       (183)          (76)      (78)
##
## Elected                                    -202         -194         -415**     -339
##                                           (339)       (796)         (171)      (271)
##
## Constant                                  913***        442          883***      367
##                                           (255)       (773)         (130)      (269)
##
## ----------------------------------------------------------------------------------
## Fixed effects for cluster?                  NO          YES            NO        YES
## Observations                               2,595       2,595         2,896      2,896
## ==================================================================================
## Note:                                                      *p<0.1; **p<0.05; ***p<0.01
```

Appendix F

Figure F.1

```
load(paste0(work_data, "projects/rewards_or_punishment/data/df_sample_manipulated.rdata"))

df_shortrun <- df %>%
  filter(election_year>1993) %>%
  filter(fractile_thres<5) %>%
  filter(t>=-6) %>%
  filter(t!=0) %>%
  filter(t<3) %>%
  as.data.frame()

### save averages for dashed lines



sr_income_pension <-
  felm(total_income_pension_euro~did + post_election + elected |0|0|cluster_id, data = df_shortrun)

sr_income <- felm(untouched_income~did + post_election + elected |0|0|cluster_id, data = df_shortrun)

avg_df <- data.frame(avg = c(sr_income_pension$beta[2], sr_income$beta[2]),
                     outcome = c("Total Income", "Earnings"))


##############################################
#### Loop split sample treatment effects #####
##############################################
na <- list()
coef <- NA
cse <- NA
covariate <- NA

#### EDUCATION COVARIATES ####
educations <- levels(df_shortrun$education)

for (i in as.character(educations)){
  na[[i]] <- felm(total_income_pension_euro~did + post_election + elected |0|0|cluster_id,
                  data = subset(df_shortrun, education==as.character(i)))
  coef[as.character(i)] <- na[[i]]$beta[2]
  cse[as.character(i)] <- na[[i]]$STATS$total_income_pension_euro$cse[2]
  covariate[paste0(i)] <- i
}


#### PARTY COVARIATES ####
df_shortrun %>%
  filter(t==1) %>%
  count(party_factor)
```

```
##             party_factor   n
## 1        socialdemocrats  92
```

```
## 2         socialliberals  18
## 3          conservatives  40
## 4       centrumdemocrats   4
## 5  socialistpeoplesparty  17
## 6        liberalalliance   3
## 7     christiandemocrats   2
## 8     danishpeoplesparty  64
## 9                liberals 108
## 10         progressparty   2
## 11        thealternative  14
```

```
# only include larger parties
parties <- c("socialdemocrats",
             "socialliberals",
             "conservatives",
             "socialistpeoplesparty",
             "danishpeoplesparty",
             "liberals",
             "thealternative")

# add to same list and vector
for (i in parties){
  na[[i]] <- felm(total_income_pension_euro~did + post_election + elected |0|0|cluster_id,
                  data = subset(df_shortrun, party_factor==as.character(i)))
  coef[as.character(i)] <- na[[i]]$beta[2]
  cse[as.character(i)] <- na[[i]]$STATS$total_income_pension_euro$cse[2]
  covariate[paste0(i)] <- i
}

#### AGE GROUP COVARIATES ####
df_shortrun %>%
  filter(t==1) %>%
  count(age_group_factor)
```

```
##   age_group_factor   n
## 1            17-29  94
## 2            30-39  83
## 3            40-49 108
## 4            50-59  65
## 5              60+  14
```

```
age_groups <- levels(df_shortrun$age_group_factor)


for (i in age_groups){
  na[[i]] <- felm(total_income_pension_euro~did + post_election + elected |0|0|cluster_id,
                  data = subset(df_shortrun, age_group_factor==as.character(i)))
  coef[as.character(i)] <- na[[i]]$beta[2]
  cse[as.character(i)] <- na[[i]]$STATS$total_income_pension_euro$cse[2]
  covariate[paste0(i)] <- i
}

# ### Treatment effects for covariates: female, age groups, right wing and majority coalition
split1 <- felm(total_income_pension_euro~did + post_election + elected |0|0|cluster_id,
               data = subset(df_shortrun, female==1))
```

```
split2 <- felm(total_income_pension_euro~did + post_election + elected |0|0|cluster_id,
               data = subset(df_shortrun, female==0))
split3 <- felm(total_income_pension_euro~did + post_election + elected |0|0|cluster_id,
               data = subset(df_shortrun, majority_coalition==1))
split4 <- felm(total_income_pension_euro~did + post_election + elected |0|0|cluster_id,
               data = subset(df_shortrun, majority_coalition==0))
split5 <- felm(total_income_pension_euro~did + post_election + elected |0|0|cluster_id,
               data = subset(df_shortrun, rightwing==1))
split6 <- felm(total_income_pension_euro~did + post_election + elected |0|0|cluster_id,
               data = subset(df_shortrun, rightwing==0))

coef[21] <- split1$beta[2]
coef[22] <- split2$beta[2]
coef[23] <- split3$beta[2]
coef[24] <- split4$beta[2]
coef[25] <- split5$beta[2]
coef[26] <- split6$beta[2]

cse[21] <- split1$STATS$total_income_pension_euro$cse[2]
cse[22] <- split2$STATS$total_income_pension_euro$cse[2]
cse[23] <- split3$STATS$total_income_pension_euro$cse[2]
cse[24] <- split4$STATS$total_income_pension_euro$cse[2]
cse[25] <- split5$STATS$total_income_pension_euro$cse[2]
cse[26] <- split6$STATS$total_income_pension_euro$cse[2]

covariate[21] <- "female"
covariate[22] <- "male"
covariate[23] <- "majority"
covariate[24] <- "minority"
covariate[25] <- "rightwing"
covariate[26] <- "leftwing"


covariates_coef_df_total <- data.frame(coef,cse,covariate) %>%
  filter(!is.na(coef)) %>% # remove the random NA
  filter(covariate!="60+") # remove 60+ candidates as their are too few observations

covariates_coef_df_total <- covariates_coef_df_total %>%
  mutate(outcome = "Total Income") %>%
  mutate(label = case_when(covariate=="primaryschool"~"Education: Primary school",
                           covariate=="highschool"~"Education: High school",
                           covariate=="vocational"~"Education: Vocational",
                           covariate=="mediumlong"~"Education: Medium-long higher",
                           covariate=="long"~"Education: Long higher",

                           covariate=="17-29"~"Age: 18-29",
                           covariate=="30-39"~"Age: 30-39",
                           covariate=="40-49"~"Age: 40-49",
                           covariate=="50-59"~"Age: 50-59",

                           covariate=="female"~"Woman",
                           covariate=="male"~"Man",
```

```
                          covariate=="rightwing"~"Right wing",
                          covariate=="leftwing"~"Left wing",
                          covariate=="majority"~"Majority coalition",
                          covariate=="minority"~"Minority coalition",

                          covariate=="socialdemocrats"~"Social Democrats",
                          covariate=="socialliberals"~"Social Liberals",
                          covariate=="conservatives"~"Conservatives",
                          covariate=="socialistpeoplesparty"~"Socialist People's Party",
                          covariate=="danishpeoplesparty"~"Danish People's Party",
                          covariate=="liberals"~"Liberals (Venstre)",
                          covariate=="thealternative"~"The Alternative")) %>%
  #ordering levels - to make order for plot
  mutate(label_factor = factor(label, levels = c("Education: Primary school",
                                                 "Education: High school",
                                                 "Education: Vocational",
                                                 "Education: Medium-long higher",
                                                 "Education: Long higher",

                                                 "Age: 18-29",
                                                 "Age: 30-39",
                                                 "Age: 40-49",
                                                 "Age: 50-59",

                                                 "Woman",
                                                 "Man",

                                                 "Right wing",
                                                 "Left wing",
                                                 "Majority coalition",
                                                 "Minority coalition",

                                                 "Social Democrats",
                                                 "Social Liberals",
                                                 "Conservatives",
                                                 "Socialist People's Party",
                                                 "Danish People's Party",
                                                 "Liberals (Venstre)",
                                                 "The Alternative")))



##############################################
#### Loop split sample treatment effects #####
##############################################
na <- list()
coef <- NA
cse <- NA
covariate <- NA

#### EDUCATION COVARIATES ####
educations <- levels(df_shortrun$education)
```

```
for (i in as.character(educations)){
  na[[i]] <- felm(untouched_income~did + post_election + elected |0|0|cluster_id,
                  data = subset(df_shortrun, education==as.character(i)))
  coef[as.character(i)] <- na[[i]]$beta[2]
  cse[as.character(i)] <- na[[i]]$STATS$untouched_income$cse[2]
  covariate[paste0(i)] <- i
}


#### PARTY COVARIATES ####
df_shortrun %>%
  filter(t==1) %>%
  count(party_factor)
```

```
##             party_factor   n
## 1        socialdemocrats  92
## 2         socialliberals  18
## 3          conservatives  40
## 4       centrumdemocrats   4
## 5  socialistpeoplesparty  17
## 6        liberalalliance   3
## 7     christiandemocrats   2
## 8     danishpeoplesparty  64
## 9               liberals 108
## 10         progressparty   2
## 11        thealternative  14
```

```
# only include larger parties
parties <- c("socialdemocrats",
             "socialliberals",
             "conservatives",
             "socialistpeoplesparty",
             "danishpeoplesparty",
             "liberals",
             "thealternative")

# add to same list and vector
for (i in parties){
  na[[i]] <- felm(untouched_income~did + post_election + elected |0|0|cluster_id,
                  data = subset(df_shortrun, party_factor==as.character(i)))
  coef[as.character(i)] <- na[[i]]$beta[2]
  cse[as.character(i)] <- na[[i]]$STATS$untouched_income$cse[2]
  covariate[paste0(i)] <- i
}

#### AGE GROUP COVARIATES ####
df_shortrun %>%
  filter(t==1) %>%
  count(age_group_factor)
```

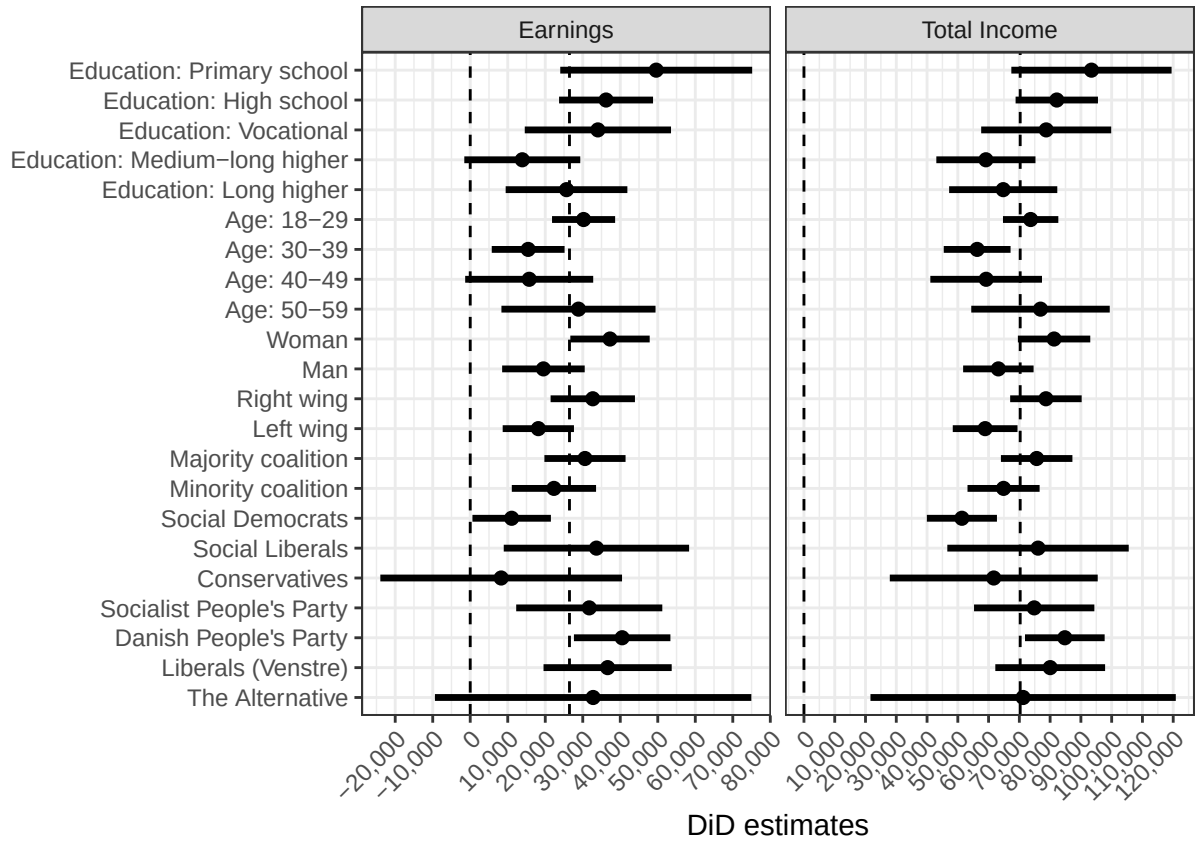
```
##   age_group_factor   n
## 1            17-29  94
## 2            30-39  83
## 3            40-49 108
```

```
## 4            50-59  65
## 5              60+  14
```

```
age_groups <- levels(df_shortrun$age_group_factor)


for (i in age_groups){
  na[[i]] <- felm(untouched_income~did + post_election + elected |0|0|cluster_id,
                  data = subset(df_shortrun, age_group_factor==as.character(i)))
  coef[as.character(i)] <- na[[i]]$beta[2]
  cse[as.character(i)] <- na[[i]]$STATS$untouched_income$cse[2]
  covariate[paste0(i)] <- i
}

# ### Treatment effects for covariates: female, age groups, right wing and majority coalition
split1 <- felm(untouched_income~did + post_election + elected |0|0|cluster_id,
               data = subset(df_shortrun, female==1))
split2 <- felm(untouched_income~did + post_election + elected |0|0|cluster_id,
               data = subset(df_shortrun, female==0))
split3 <- felm(untouched_income~did + post_election + elected |0|0|cluster_id,
               data = subset(df_shortrun, majority_coalition==1))
split4 <- felm(untouched_income~did + post_election + elected |0|0|cluster_id,
               data = subset(df_shortrun, majority_coalition==0))
split5 <- felm(untouched_income~did + post_election + elected |0|0|cluster_id,
               data = subset(df_shortrun, rightwing==1))
split6 <- felm(untouched_income~did + post_election + elected |0|0|cluster_id,
               data = subset(df_shortrun, rightwing==0))

coef[21] <- split1$beta[2]
coef[22] <- split2$beta[2]
coef[23] <- split3$beta[2]
coef[24] <- split4$beta[2]
coef[25] <- split5$beta[2]
coef[26] <- split6$beta[2]

cse[21] <- split1$STATS$untouched_income$cse[2]
cse[22] <- split2$STATS$untouched_income$cse[2]
cse[23] <- split3$STATS$untouched_income$cse[2]
cse[24] <- split4$STATS$untouched_income$cse[2]
cse[25] <- split5$STATS$untouched_income$cse[2]
cse[26] <- split6$STATS$untouched_income$cse[2]

covariate[21] <- "female"
covariate[22] <- "male"
covariate[23] <- "majority"
covariate[24] <- "minority"
covariate[25] <- "rightwing"
covariate[26] <- "leftwing"


covariates_coef_df_earnings <- data.frame(coef,cse,covariate) %>%
  filter(!is.na(coef)) %>% # remove the random NA
  filter(covariate!="60+") # remove 60+ candidates as their are too few observations
```

```r
covariates_coef_df_earnings <- covariates_coef_df_earnings %>%
  mutate(outcome = "Earnings") %>%
  mutate(label = case_when(covariate=="primaryschool"~"Education: Primary school",
                           covariate=="highschool"~"Education: High school",
                           covariate=="vocational"~"Education: Vocational",
                           covariate=="mediumlong"~"Education: Medium-long higher",
                           covariate=="long"~"Education: Long higher",

                           covariate=="17-29"~"Age: 18-29",
                           covariate=="30-39"~"Age: 30-39",
                           covariate=="40-49"~"Age: 40-49",
                           covariate=="50-59"~"Age: 50-59",

                           covariate=="female"~"Woman",
                           covariate=="male"~"Man",

                           covariate=="rightwing"~"Right wing",
                           covariate=="leftwing"~"Left wing",
                           covariate=="majority"~"Majority coalition",
                           covariate=="minority"~"Minority coalition",

                           covariate=="socialdemocrats"~"Social Democrats",
                           covariate=="socialliberals"~"Social Liberals",
                           covariate=="conservatives"~"Conservatives",
                           covariate=="socialistpeoplesparty"~"Socialist People's Party",
                           covariate=="danishpeoplesparty"~"Danish People's Party",
                           covariate=="liberals"~"Liberals (Venstre)",
                           covariate=="thealternative"~"The Alternative")) %>%
  #ordering levels - to make order for plot
  mutate(label_factor = factor(label, levels = c("Education: Primary school",
                                                 "Education: High school",
                                                 "Education: Vocational",
                                                 "Education: Medium-long higher",
                                                 "Education: Long higher",

                                                 "Age: 18-29",
                                                 "Age: 30-39",
                                                 "Age: 40-49",
                                                 "Age: 50-59",

                                                 "Woman",
                                                 "Man",

                                                 "Right wing",
                                                 "Left wing",
                                                 "Majority coalition",
                                                 "Minority coalition",

                                                 "Social Democrats",
                                                 "Social Liberals",
                                                 "Conservatives",
                                                 "Socialist People's Party",
                                                 "Danish People's Party",
```

```
                                                "Liberals (Venstre)",
                                                "The Alternative")))

covariates_coef_df <- bind_rows(covariates_coef_df_total, covariates_coef_df_earnings)

covariates_coef_df %>%
  ggplot(data =., aes(x = label_factor, y = coef)) +
  geom_point(col = "black",
             size = 2) +
  geom_hline(yintercept = 0, linetype="dashed") +
  geom_hline(aes(yintercept = avg), data = avg_df, color = "black", linetype = "dashed") +
  geom_linerange(aes(ymin = coef-(cse*1.96),
                     ymax = coef+(cse*1.96)),
                 col = "black",
                 size = 1.2) +
  theme_bw() +
  xlab("") +
  ylab("DiD estimates") +
  theme(axis.text.x = element_text(angle=45,hjust = 1)) +
  coord_flip() +
  scale_x_discrete(limits = rev(levels(covariates_coef_df$label_factor))) +
  scale_y_continuous(breaks = seq(-20000,130000,10000), label = scales::comma) +
  facet_grid(~outcome, scales = "free")
```

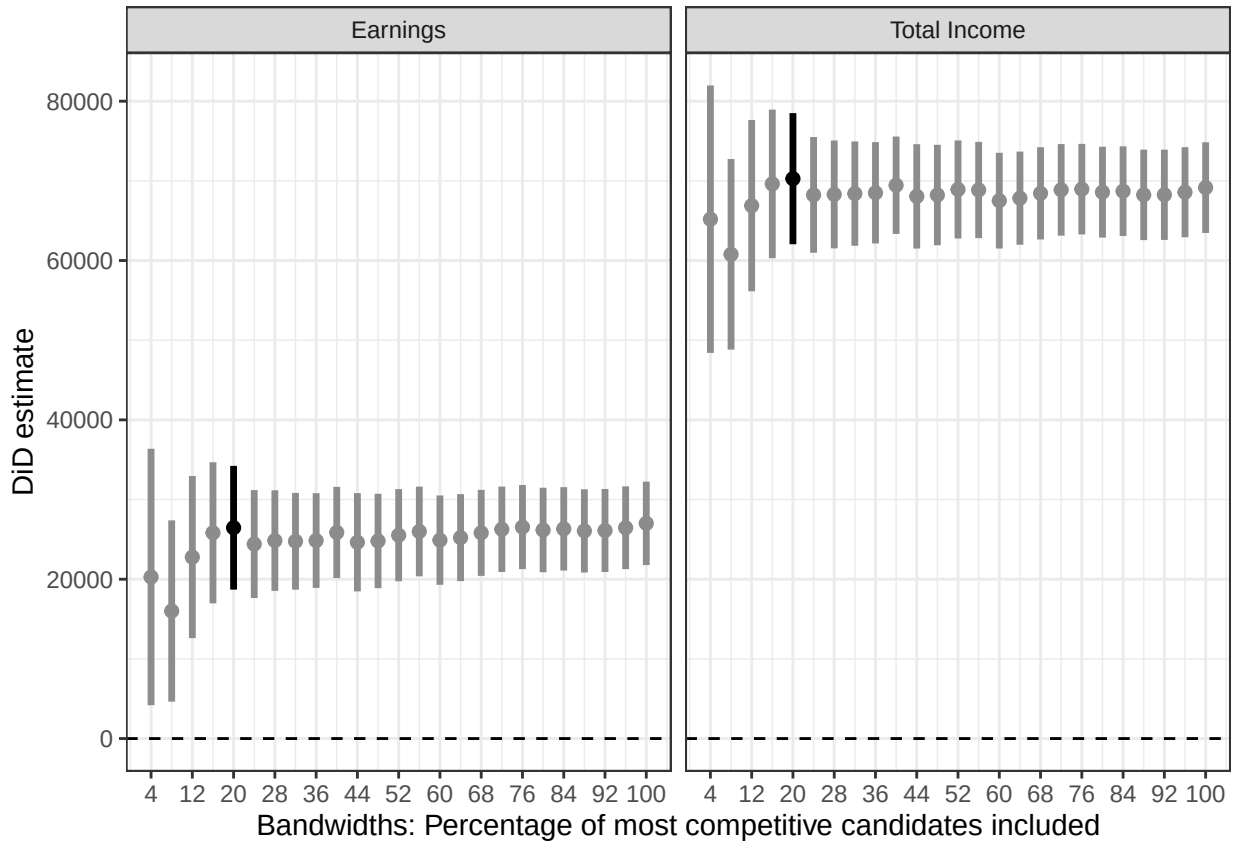


Appendix G

Figure G.1

```
load(paste0(work_data, "projects/rewards_or_punishment/data/df_sample_manipulated.rdata"))

df_shortrun <- df %>%
  filter(election_year>1993) %>%
  filter(t!=0) %>%
  filter(t<3)

na1 <- NA
coef1 <- NA
cse1 <- NA

for (i in 1:25){
  na1 <- felm(total_income_pension_euro~did + post_election + elected |0|0|cluster_id,
              data = subset(df_shortrun, fractile_thres<i))
  coef1[i] <- na1$beta[2]
  cse1[i] <- na1$STATS$total_income_pension_euro$cse[2]
}

bw_df_total <- data.frame(coef1,cse1)

bw_df_total <- bw_df_total %>%
  dplyr::mutate(outcome = "Total Income") %>%
  rename(coef = coef1,
         cse = cse1)


na2 <- NA
coef2 <- NA
cse2 <- NA


for (i in 1:25){
  na2 <- felm(untouched_income~did + post_election + elected |0|0|cluster_id,
              data = subset(df_shortrun, fractile_thres<i))
  coef2[i] <- na2$beta[2]
  cse2[i] <- na2$STATS$untouched_income$cse[2]
}

bw_df_earnings <- data.frame(coef2,cse2)

bw_df_earnings <- bw_df_earnings %>%
  dplyr::mutate(outcome = "Earnings") %>%
  rename(coef = coef2,
         cse = cse2)


bw_df <- bind_rows(bw_df_earnings, bw_df_total)

bw_df$fractile_thres <- seq(0:24)
```

```r
bw_df <- bw_df %>%
  dplyr::mutate(closeness = fractile_thres*4) %>%
  mutate(optimal = ifelse(closeness==20,"Optimal", "BW"))

ggplot(bw_df, aes(x = closeness, y = coef, color = optimal)) +
  geom_point(size = 2) +
  geom_hline(yintercept = 0, linetype="dashed") +
  geom_linerange(aes(ymin = coef-(cse*1.96),
                     ymax = coef+(cse*1.96)),
                 size = 1.2) +
  scale_color_manual(values = c("grey55","black")) +
  theme_bw() +
  theme(legend.position = "none") +
  labs(x="Bandwidths: Percentage of most competitive candidates included", y="DiD estimate") +
  scale_x_continuous(breaks = seq(4,100,8), labels = seq(4,100,8)) +
  facet_wrap(outcome~., ncol=2)
```

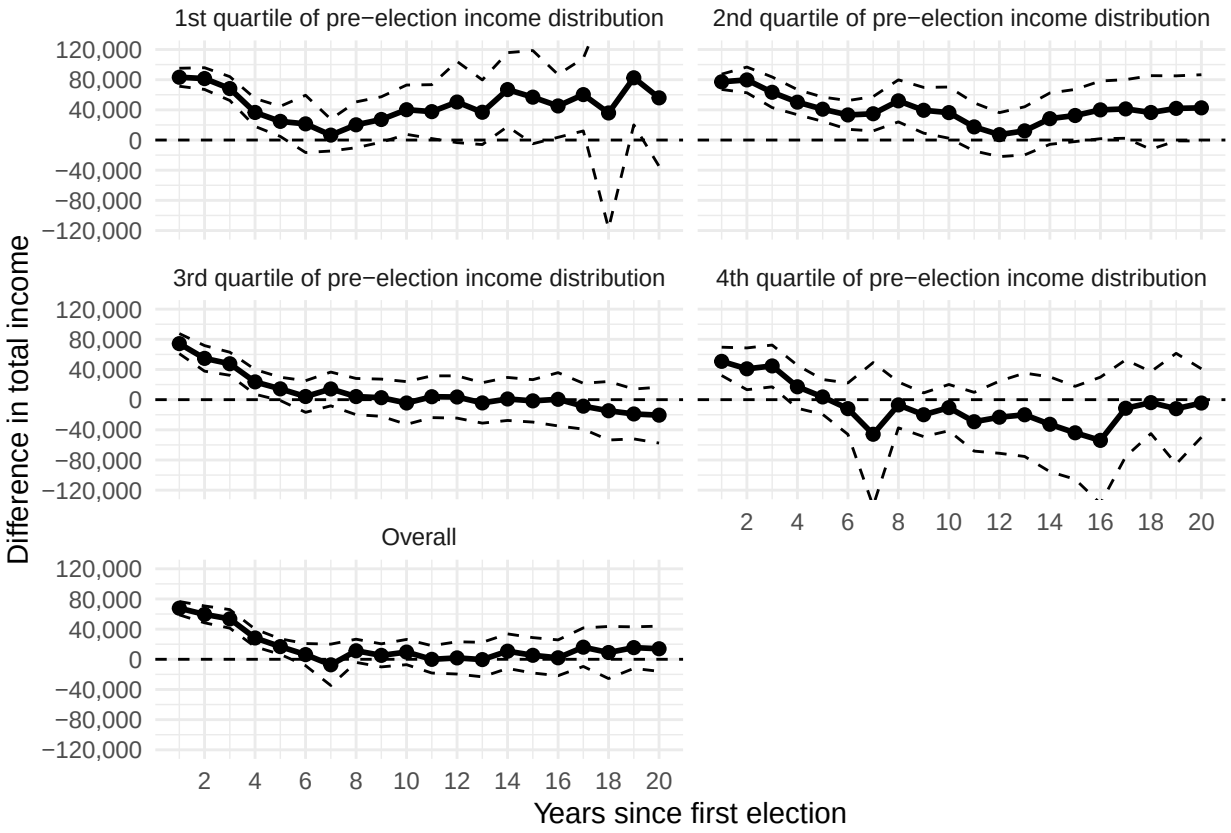


Appendix H

Figure H.1

```
#load(paste0(work_data, "projects/rewards_or_punishment/data/df_longrun_manipulated.rdata"))
load(paste0(work_data, "projects/rewards_or_punishment/data/df_sample_manipulated.rdata"))

df_longrun <- df %>%
  filter(election_year>1993) %>%
  filter(fractile_thres<5)

## estimate return for every t

lr_income <- df_longrun %>%
  filter(t>0 & t<21) %>%
  group_by(t) %>%
  do(tidy(felm(total_income_pension_euro~elected |0|0|cluster_id, data=.))) %>%
  filter(term=="elected")

lr_income <- lr_income %>%
  mutate(lower = estimate-(std.error*1.96),
         upper = estimate+(std.error*1.96))


## winsorize income and look at income developments across pre-election quartiles
lr_income_quantiles_wins <- df_longrun %>%
  filter(t>0 & t<21) %>%
  group_by(t, income_pre_percentile) %>%
  do(tidy(lm(total_income_pension_euro~elected, data=.))) %>%
  filter(term=="elected" & !is.na(income_pre_percentile))

# join overall long run df with quartile df
# add variable quantile to overall df
lr_income$income_pre_percentile <- "Overall"

lr_income_quantiles_wins <- lr_income_quantiles_wins %>%
  mutate(income_pre_percentile =
           case_when(income_pre_percentile==1~"1st quartile of pre-election income distribution",
                     income_pre_percentile==2~"2nd quartile of pre-election income distribution",
                     income_pre_percentile==3~"3rd quartile of pre-election income distribution",
                     income_pre_percentile==4~"4th quartile of pre-election income distribution")) %>%
  mutate(income_pre_percentile = factor(income_pre_percentile,
                                        levels = c("1st quartile of pre-election income distribution",
                                        "2nd quartile of pre-election income distribution",
                                        "3rd quartile of pre-election income distribution",
                                        "4th quartile of pre-election income distribution")))

# bind estimate-dfs together - quartile and overall
plot_lr <- bind_rows(lr_income, lr_income_quantiles_wins)

plot_lr <- plot_lr %>%
  mutate(lower = estimate-(std.error*1.96),
         upper = estimate+(std.error*1.96))
```

```
ggplot(plot_lr, aes(x = t, y = estimate, color = factor(income_pre_percentile))) +
  geom_point(size = 2, color = "black") +
  geom_line(size=1) +
  geom_line(aes(y = lower), linetype = 2) +
  geom_line(aes(y = upper), linetype = 2) +
  geom_hline(yintercept = 0, linetype="dashed") +
  theme_minimal() +
  labs(x="Years since first election", y="Difference in total income") +
  scale_x_continuous(breaks = seq(2,20,2), labels = seq(2,20,2)) +
  coord_cartesian(ylim = c(-120000,120000)) +
  scale_y_continuous(breaks = seq(-120000,120000,40000), label = scales::comma) +
  scale_color_grey(name="Income quartile \n(pre election)", start = 0, end = 0) +
  theme(legend.position = "none") +
  facet_wrap(. ~income_pre_percentile, ncol=2)
```



Appendix I

Figure I.1

```
load(paste0(work_data, "projects/rewards_or_punishment/replication/candidates_list.rdata"))

all_candidates <- bind_rows(candidates_list)

# select the relevant sample
career_df <- all_candidates %>%
  filter(election_year>1991 & election_year<2019 & first_run==1 &
         fractile_thres<5 & OPSTILLINGSFORM=="Sideordnet")

# make dataframe long
career_df_long <- career_df %>%
  dplyr::select(PNR, elected, election_year, incumbent_future, incumbent_future_2,
                incumbent_future_3) %>%
  mutate(incumbent_0 = elected) %>%
  dplyr::rename(incumbent_1 = incumbent_future,
         incumbent_2 = incumbent_future_2,
         incumbent_3 = incumbent_future_3) %>%
  tidyr::pivot_longer(cols = starts_with("incumbent"),
                      names_to = c(".value", "time"),
                      names_sep = "_")

# measure share of incumbents over time for initial winners and losers
df_agg <- career_df_long %>%
  group_by(time, elected) %>%
  summarise(mean_inc = mean(incumbent))

df_agg %>%
  mutate(factor_elected = as.factor(ifelse(elected==1,"Yes","No"))) %>%
  ggplot(data=., aes(x=time, y = mean_inc,  fill = factor_elected, color = factor_elected)) +
  geom_col(aes(fill = factor_elected, color = factor_elected), position = position_dodge()) +
  scale_fill_grey(name="Elected in year t0", start = 0.75, end = 0.35) +
  scale_color_grey(name="Elected in year t0", start = 0.85, end = 0.25) +
  scale_y_continuous(labels = seq(0,1,0.1), breaks = seq(0,1,0.1)) +
  xlab("Election, t") +
  ylab("Share of incumbents") +
  theme_minimal()
```

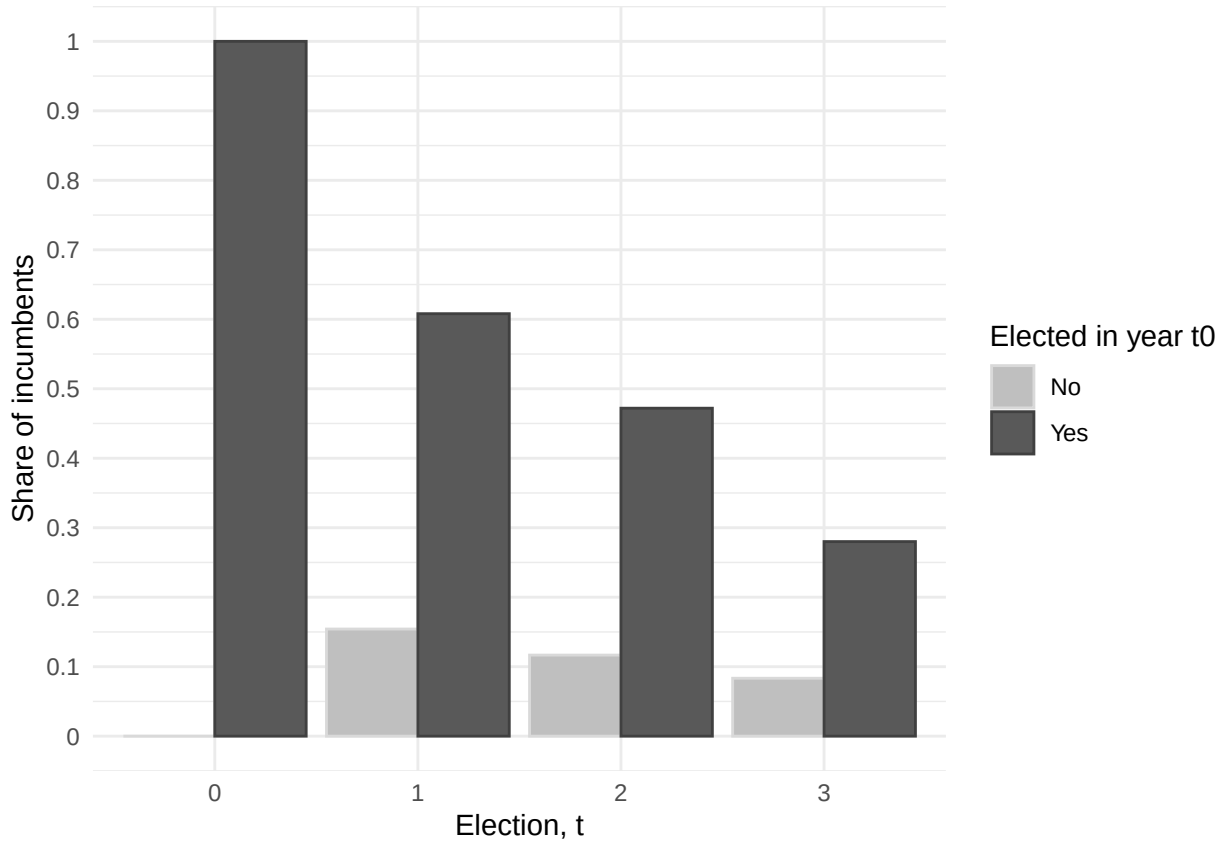


Figure I.2

```
load(paste0(work_data, "projects/rewards_or_punishment/data/df_sample_manipulated.rdata"))

df_pt <- df %>%
  filter(election_year>1993) %>%
  filter(fractile_thres<5) %>%
  filter(t>=-6) %>%
  filter(t!=0) %>%
  filter(t<3) %>%
  mutate(decade = case_when(election_year %in% c(1994,1998)~"90s",
                   election_year %in% c(2001,2005,2007)~"00s",
                   election_year %in% c(2011,2015)~"10s"))

plot_pt_naive <- df_pt %>%
  group_by(factor_elected, t, decade) %>%
  summarise(mean_income = mean(untouched_income, na.rm=TRUE),
            std_error = sqrt(var(untouched_income)/length(untouched_income)-1)) %>%
  mutate(outcome = "Earnings")


plot_pt_allowance_pension <- df_pt %>%
  group_by(factor_elected, t, decade) %>%
  summarise(mean_income = mean(total_income_pension_euro, na.rm=TRUE),
            std_error = sqrt(var(total_income_pension_euro)/length(total_income_pension_euro)-1)) %>%
```

```
  mutate(outcome = "Total Income")

plot_pt <- bind_rows(plot_pt_naive, plot_pt_allowance_pension) %>%
  mutate(outcome = factor(outcome, levels = c("Earnings", "Total Income")),
         decade = factor(decade, levels =c("90s","00s","10s")))

plot1 <-
  plot_pt %>%
  filter(t<3) %>%
  filter(outcome!="Earnings + Allowance") %>%
  ggplot(data=., aes(x=t, y=mean_income, color = factor_elected, shape = factor_elected)) +
  geom_point(position = position_dodge(width = 0.6), size = 2.5) +
  geom_linerange(position = position_dodge(width = 0.6),
                 aes(ymin=mean_income-(std_error*1.96),
                     ymax=mean_income+(std_error*1.96)),
                 size = 0.7) +
  geom_vline(xintercept = 0, linetype = "longdash") +
  scale_x_continuous(breaks = seq(-6,3,1), labels = seq(-6,3,1)) +
  scale_y_continuous(breaks = seq(0,160000,20000), labels = scales::comma) +
  xlab("") +
  ylab("Euros") +
  labs(color="Elected", shape="Elected") +
  scale_color_manual(values = c("black", "grey55")) +
  scale_alpha_manual(values = c(0.2, 0.5, 0.8)) +
  theme_bw() +
  coord_cartesian(ylim = c(0,160000)) +
  facet_wrap(~decade+outcome,ncol = 2)

plot1
```

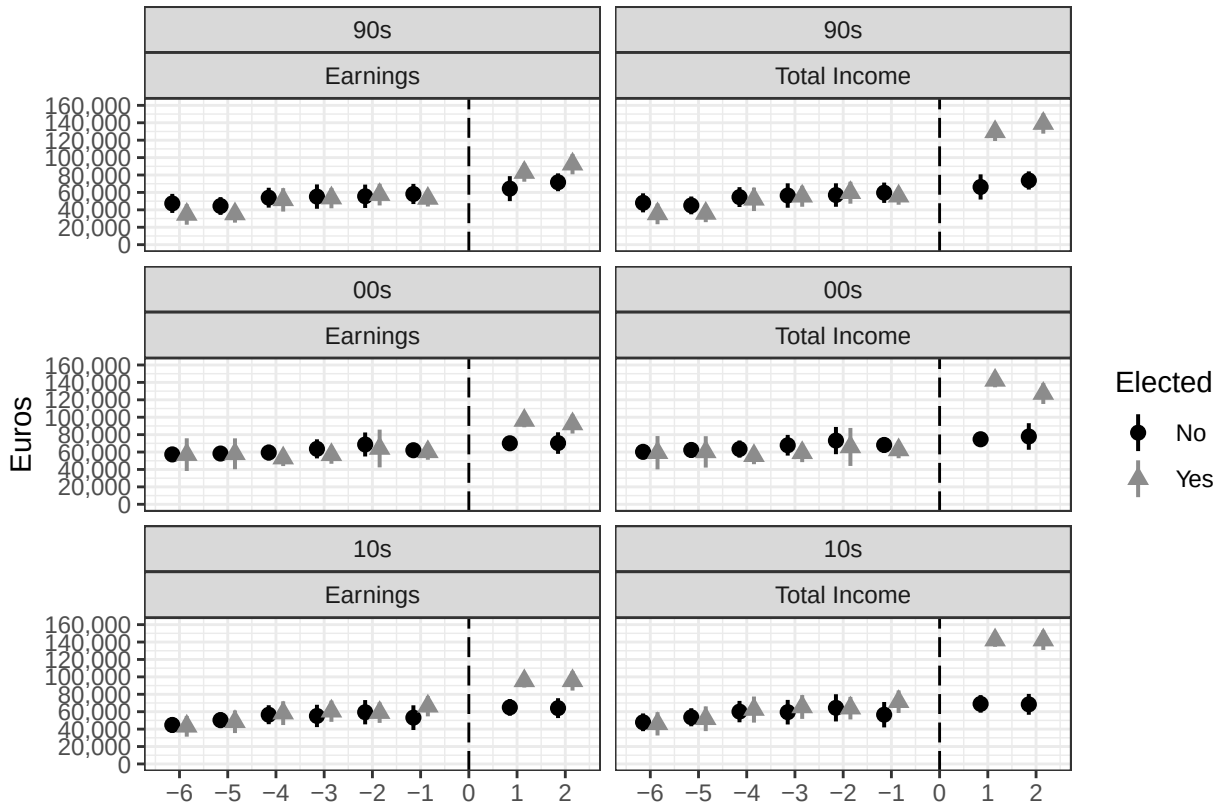


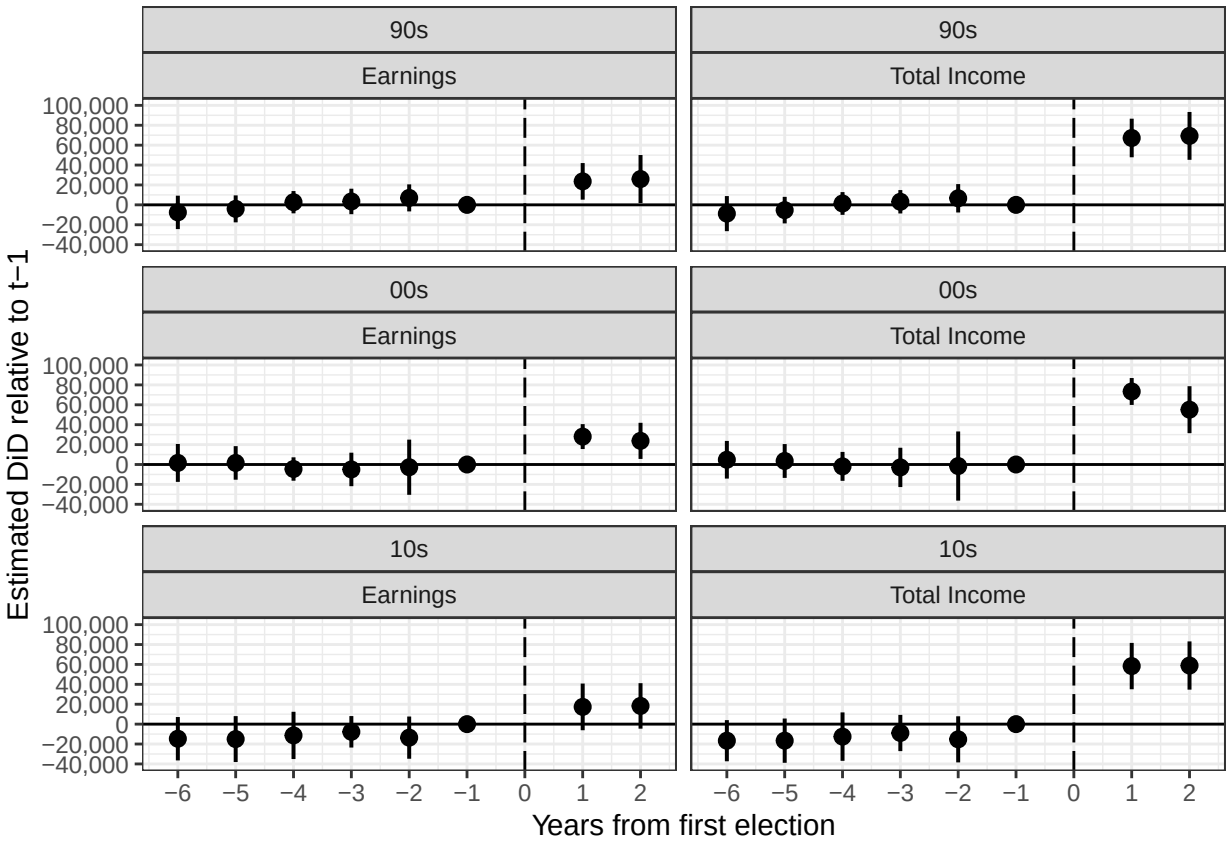
Figure I.3

```
#### annual atts
df_att <- df_pt %>%
  mutate(t_factor = factor(t, levels = c(-1,-3,-2, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
                                         12, 13, 14, 15, 16)),
         pnr_num = as.numeric(PNR))

df_att_90 <- df_att %>% filter(decade=="90s")
df_att_00 <- df_att %>% filter(decade=="00s")
df_att_10 <- df_att %>% filter(decade=="10s")

att_naive_reg_90 <- did::att_gt(yname = "untouched_income",
                                tname = "t",
                                id = "pnr_num",
                                gname = "elected",
                                est_method = "dr",
                                control_group = "nevertreated",
                                anticipation = 0,
                                panel = TRUE,
                                clustervars = "cluster_id",
                                bstrap = TRUE,
                                biters = 1000,
                                base_period = "universal",
                                allow_unbalanced_panel = TRUE,
```

```
                              pl = TRUE,
                              cores = 6,
                              data = df_att_90)

att_naive_90 <- data.frame(outcome = "Earnings",
                           decade = "90s",
                           t = att_naive_reg_90$t,
                           att = att_naive_reg_90$att,
                           se = att_naive_reg_90$se)

att_total_reg_90 <- did::att_gt(yname = "total_income_pension_euro",
                                tname = "t",
                                id = "pnr_num",
                                gname = "elected",
                                est_method = "dr",
                                control_group = "nevertreated",
                                anticipation = 0,
                                panel = TRUE,
                                clustervars = "cluster_id",
                                bstrap = TRUE,
                                biters = 1000,
                                base_period = "universal",
                                allow_unbalanced_panel = TRUE,
                                pl = TRUE,
                                cores = 6,
                                data = df_att_90)

att_total_90 <- data.frame(outcome = "Total Income",
                           decade = "90s",
                           t = att_total_reg_90$t,
                           att = att_total_reg_90$att,
                           se = att_total_reg_90$se)



### 00s
att_naive_reg_00 <- did::att_gt(yname = "untouched_income",
                                tname = "t",
                                id = "pnr_num",
                                gname = "elected",
                                est_method = "dr",
                                control_group = "nevertreated",
                                anticipation = 0,
                                panel = TRUE,
                                clustervars = "cluster_id",
                                bstrap = TRUE,
                                biters = 1000,
                                base_period = "universal",
                                allow_unbalanced_panel = TRUE,
                                pl = TRUE,
                                cores = 6,
                                data = df_att_00)
```

```r
att_naive_00 <- data.frame(outcome = "Earnings",
                           decade = "00s",
                           t = att_naive_reg_00$t,
                           att = att_naive_reg_00$att,
                           se = att_naive_reg_00$se)

att_total_reg_00 <- did::att_gt(yname = "total_income_pension_euro",
                                tname = "t",
                                id = "pnr_num",
                                gname = "elected",
                                est_method = "dr",
                                control_group = "nevertreated",
                                anticipation = 0,
                                panel = TRUE,
                                clustervars = "cluster_id",
                                bstrap = TRUE,
                                biters = 1000,
                                base_period = "universal",
                                allow_unbalanced_panel = TRUE,
                                pl = TRUE,
                                cores = 6,
                                data = df_att_00)

att_total_00 <- data.frame(outcome = "Total Income",
                           decade = "00s",
                           t = att_total_reg_00$t,
                           att = att_total_reg_00$att,
                           se = att_total_reg_00$se)

### 10s
att_naive_reg_10 <- did::att_gt(yname = "untouched_income",
                                tname = "t",
                                id = "pnr_num",
                                gname = "elected",
                                est_method = "dr",
                                control_group = "nevertreated",
                                anticipation = 0,
                                panel = TRUE,
                                clustervars = "cluster_id",
                                bstrap = TRUE,
                                biters = 1000,
                                base_period = "universal",
                                allow_unbalanced_panel = TRUE,
                                pl = TRUE,
                                cores = 6,
                                data = df_att_10)

att_naive_10 <- data.frame(outcome = "Earnings",
                           decade = "10s",
                           t = att_naive_reg_10$t,
                           att = att_naive_reg_10$att,
                           se = att_naive_reg_10$se)
```

```r
att_total_reg_10 <- did::att_gt(yname = "total_income_pension_euro",
                                tname = "t",
                                id = "pnr_num",
                                gname = "elected",
                                est_method = "dr",
                                control_group = "nevertreated",
                                anticipation = 0,
                                panel = TRUE,
                                clustervars = "cluster_id",
                                bstrap = TRUE,
                                biters = 1000,
                                base_period = "universal",
                                allow_unbalanced_panel = TRUE,
                                pl = TRUE,
                                cores = 6,
                                data = df_att_10)

att_total_10 <- data.frame(outcome = "Total Income",
                           decade = "10s",
                           t = att_total_reg_10$t,
                           att = att_total_reg_10$att,
                           se = att_total_reg_10$se)


att_decade_df <- bind_rows(att_naive_90, att_total_90, att_naive_00, att_total_00,
                           att_naive_10, att_total_10) %>%
  mutate(outcome = factor(outcome, levels = c("Earnings", "Total Income")),
         decade = factor(decade, levels =c("90s","00s","10s")))


plot2 <-
  att_decade_df %>%
  filter(t<3) %>%
  ggplot(data=., aes(x=t, y=att)) +
  geom_point(position = position_dodge(width = 0.4), size = 2.5) +
  geom_linerange(position = position_dodge(width = 0.4),
                 aes(ymin=att-(se*1.96),
                     ymax=att+(se*1.96)),
                 size = 0.7) +
  geom_vline(xintercept = 0, linetype = "longdash") +
  geom_hline(yintercept = 0, linetype = "solid") +
  scale_x_continuous(breaks = seq(-6,3,1), labels = seq(-6,3,1)) +
  scale_y_continuous(breaks = seq(-40000,160000,20000), labels = scales::comma) +
  xlab("Years from first election") +
  ylab("Estimated DiD relative to t-1") +
  theme_bw() +
  coord_cartesian(ylim = c(-40000,100000)) +
  theme(legend.position = c(0.25,0.85), legend.title = element_blank(),
        legend.box.background = element_rect(color = "black")) +
  facet_wrap(~decade+outcome,ncol=2)

plot2
```


90s
Earnings
90s
Total Income
00s
Earnings
00s
Total Income
10s
Earnings
10s
Total Income
Estimated DiD relative to t−1
100,000
80,000
60,000
40,000
20,000
0
−20,000
−40,000
−6
−5
−4
−3
−2
−1
0
1
2
Years from first election

Appendix J

Figure J.1

```
load(paste0(work_data, "projects/rewards_or_punishment/data/df_sample_manipulated.rdata"))

df_longrun <- df %>%
  filter(election_year>1993) %>%
  filter(fractile_thres<5) %>%
  filter(t>0)

# create age group based on t-1 and merge it to df
df_agegroup <- df %>%
  filter(election_year>1993) %>%
  filter(fractile_thres<5) %>%
  filter(t==-1) %>%
  mutate(age_third = ntile(age,3)) %>%
  dplyr::select(c(PNR, age_third))


## Age cut-offs are 33 years and 46 years
df_longrun <- df_longrun %>%
  left_join(df_agegroup)


# one candidate did not appear in the data in t-1 and therefore their age_third must be added manually
# they were 33 in t+1 and hence 31 in t-1. Therefore they are assigned age_third 1

df_longrun <- df_longrun %>%
  mutate(age_third = replace_na(age_third, 1))


######################
# create npv variables, for rate of 1,24 and 2,58 and for income inclusive/exclusive pension

### Calculate net present value
df_longrun <- df_longrun %>%
  dplyr::mutate(npv_124 = total_income_euro/(1+0.0124)^t,
                npv_258 = total_income_euro/(1+0.0258)^t,
                npv_pens_124 = total_income_pension_euro/(1+0.0124)^t,
                npv_pens_258 = total_income_pension_euro/(1+0.0258)^t)

# make cumulative net present value
df_longrun <- df_longrun %>%
  group_by(PNR) %>%
  dplyr::mutate(npv_124_cumu = cumsum(npv_124),
                npv_258_cumu = cumsum(npv_258),
                npv_pens_124_cumu = cumsum(npv_pens_124),
                npv_pens_258_cumu = cumsum(npv_pens_258))


#### CALCULATE DIF IN MEANS ACROSS T

# calculate dif-in-means, create dfs of estimates and ad variables for labels etc
```

```r
npv_124_cumu <- df_longrun %>%
  filter(t>0 & t<21) %>%
  group_by(t) %>%
  do(tidy(lm_robust(npv_124_cumu~elected, data=., clusters = cluster_id))) %>%
  filter(term=="elected")

npv_pens_124_cumu <- df_longrun %>%
  filter(t>0 & t<21) %>%
  group_by(t) %>%
  do(tidy(lm_robust(npv_pens_124_cumu~elected, data=., clusters = cluster_id))) %>%
  filter(term=="elected")

npv_258_cumu <- df_longrun %>%
  filter(t>0 & t<21) %>%
  group_by(t) %>%
  do(tidy(lm_robust(npv_258_cumu~elected, data=., clusters = cluster_id))) %>%
  filter(term=="elected")

npv_pens_258_cumu <- df_longrun %>%
  filter(t>0 & t<21) %>%
  group_by(t) %>%
  do(tidy(lm_robust(npv_pens_258_cumu~elected, data=., clusters = cluster_id))) %>%
  filter(term=="elected")

npv_124_cumu$rate <- "Interest rate of 1.24 percent"
npv_pens_124_cumu$rate <- "Interest rate of 1.24 percent"
npv_258_cumu$rate <- "Interest rate of 2.58 percent"
npv_pens_258_cumu$rate <- "Interest rate of 2.58 percent"

npv_124_cumu$outcome <- "Income excl. pension"
npv_258_cumu$outcome <- "Income excl. pension"
npv_pens_124_cumu$outcome <- "Income incl. pension"
npv_pens_258_cumu$outcome <- "Income incl. pension"

plot_df <- bind_rows(npv_124_cumu, npv_258_cumu, npv_pens_124_cumu, npv_pens_258_cumu)

plot_df <- plot_df %>%
  dplyr::mutate(outcome = factor(outcome, levels = c("Income incl. pension", "Income excl. pension")))


################# 2.58 RATE INCLUDING PENSION
plot_df %>%
  filter(rate=="Interest rate of 2.58 percent") %>%
  filter(outcome=="Income incl. pension") %>%
  ggplot(data=., aes(x = t, y = estimate, scale = outcome)) +
  geom_point(size = 2, aes(shape = outcome)) +
  geom_line(size=1) +
  geom_line(aes(y = conf.low), linetype=2) +
  geom_line(aes(y = conf.high), linetype=2) +
  geom_hline(yintercept = 0, linetype=1) +
  theme_minimal() +
  theme(legend.position="none") +
  scale_fill_grey(name="", start = 0.75, end = 0.35) +
```

```
  scale_color_grey(name="", start = 0.75, end = 0.35) +
  scale_shape_manual(name="", values =c(19,19,17,17)) +
  guides(alpha="none")  +
  scale_x_continuous(breaks = seq(2,20,2), labels = seq(2,20,2)) +
  coord_cartesian(ylim = c(-100000,700000)) +
  scale_y_continuous(breaks = seq(-100000,700000,100000), label = scales::comma) +
  labs(x="Years from first election", y="Difference in cumulative net present value of total income")
```

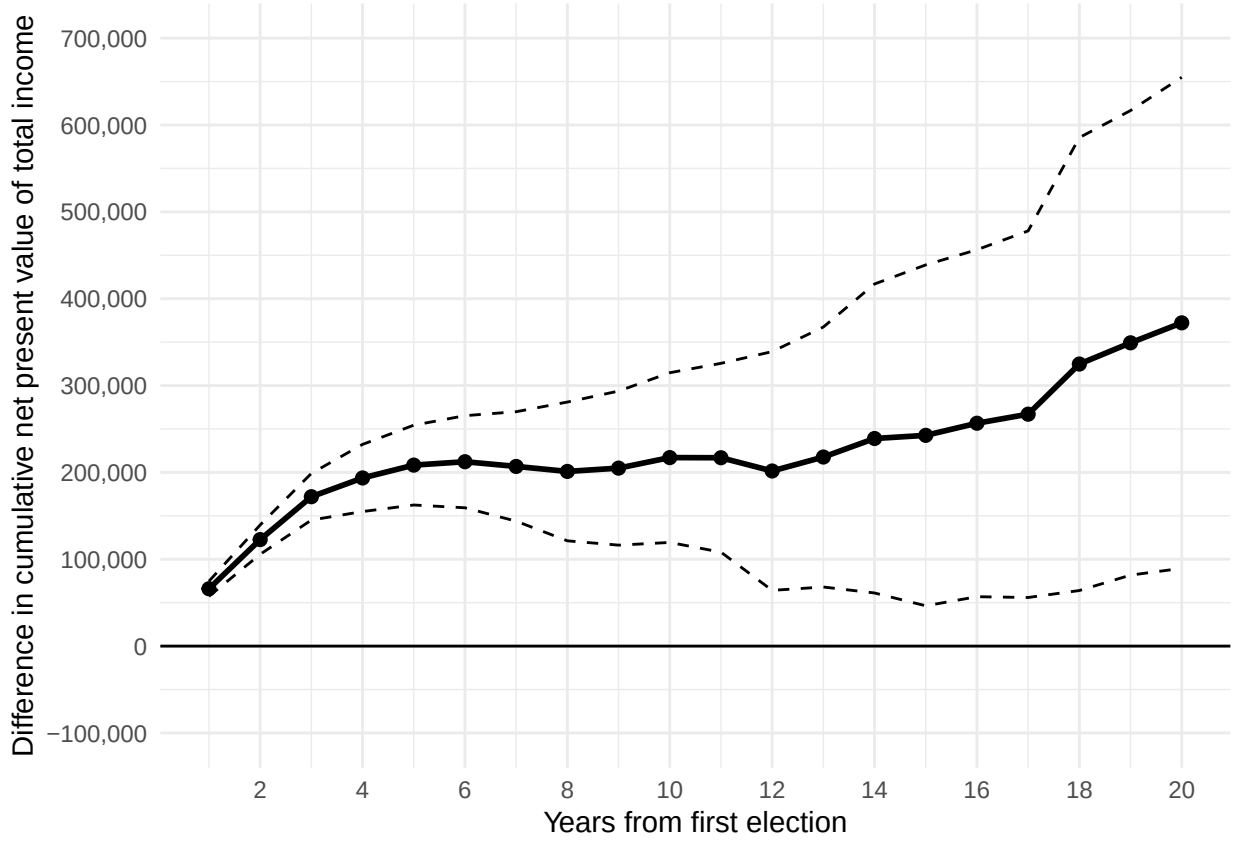


Figure J.2

```
#### Rate of 2.58 EXCLUSIVE PENSION
plot_df %>%
  filter(rate=="Interest rate of 2.58 percent") %>%
  filter(outcome=="Income excl. pension") %>%
  ggplot(data=., aes(x = t, y = estimate, scale = outcome)) +
  geom_point(size = 2, aes(shape = outcome)) +
  geom_line(size=1) +
  geom_line(aes(y = conf.low), linetype=2) +
  geom_line(aes(y = conf.high), linetype=2) +
  geom_hline(yintercept = 0, linetype=1) +
  theme_minimal() +
  theme(legend.position="none") +
  scale_fill_grey(name="", start = 0.75, end = 0.35) +
  scale_color_grey(name="", start = 0.75, end = 0.35) +
  scale_shape_manual(name="", values =c(19,19,17,17)) +
```

```
  guides(alpha="none")  +
  scale_x_continuous(breaks = seq(2,20,2), labels = seq(2,20,2)) +
  coord_cartesian(ylim = c(-100000,700000)) +
  scale_y_continuous(breaks = seq(-100000,700000,100000), label = scales::comma) +
  labs(x="Years since first election", y="Difference in cumulative net present value of income")
```

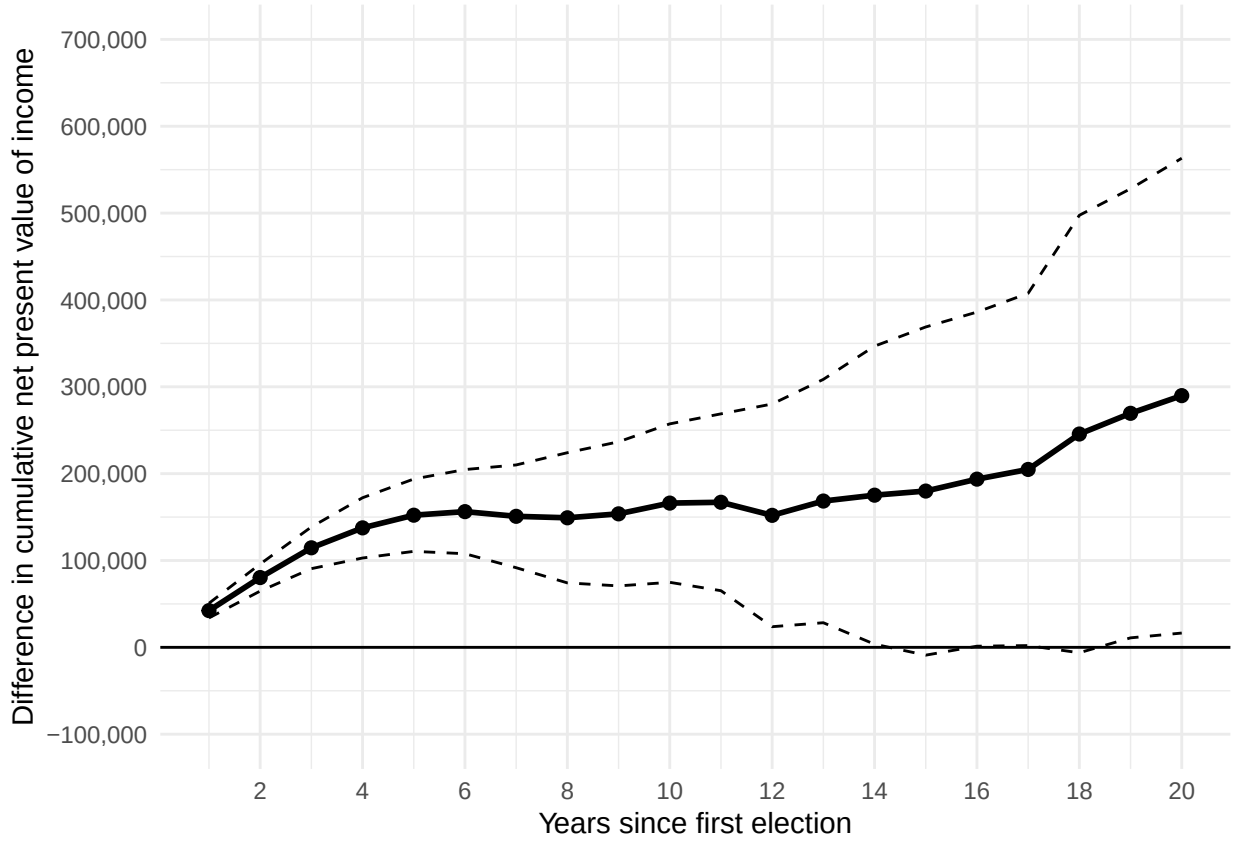


Figure J.3

```
plot_df %>%
  filter(rate=="Interest rate of 1.24 percent") %>%
  ggplot(data=., aes(x = t, y = estimate, scale = outcome)) +
  geom_point(size = 2, aes(shape = outcome)) +
  geom_line(size=1) +
  geom_line(aes(y = conf.low), linetype=2) +
  geom_line(aes(y = conf.high), linetype=2) +
  geom_hline(yintercept = 0, linetype=1) +
  theme_minimal() +
  theme(legend.position="none") +
  facet_wrap(~outcome, ncol=2) +
  scale_fill_grey(name="", start = 0.75, end = 0.35) +
  scale_color_grey(name="", start = 0.75, end = 0.35) +
  scale_shape_manual(name="", values =c(19,19,17,17)) +
  guides(alpha="none")  +
  scale_x_continuous(breaks = seq(2,20,2), labels = seq(2,20,2)) +
  coord_cartesian(ylim = c(-100000,700000)) +
```

```
  scale_y_continuous(breaks = seq(-100000,700000,100000), label = scales::comma) +
  labs(x="Years from first election", y="Difference in cumulative net present value of total income")
```

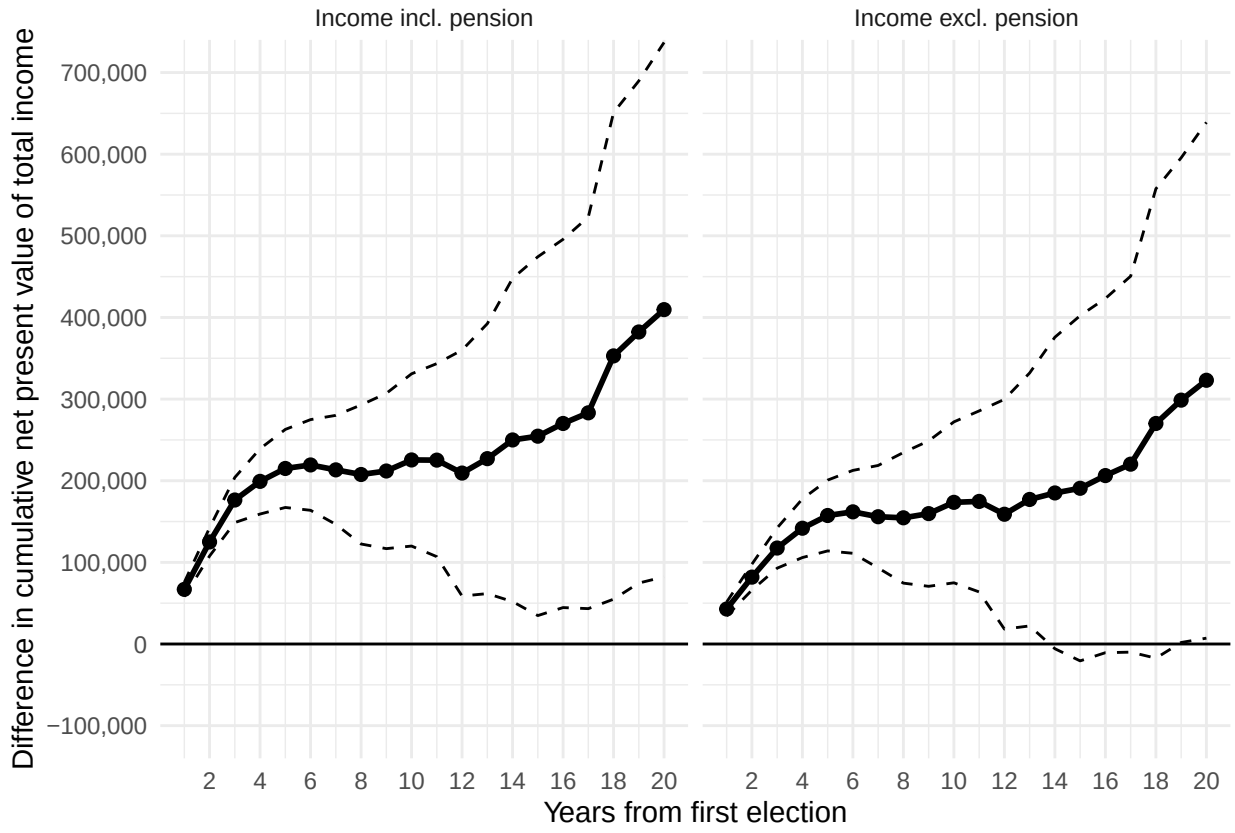
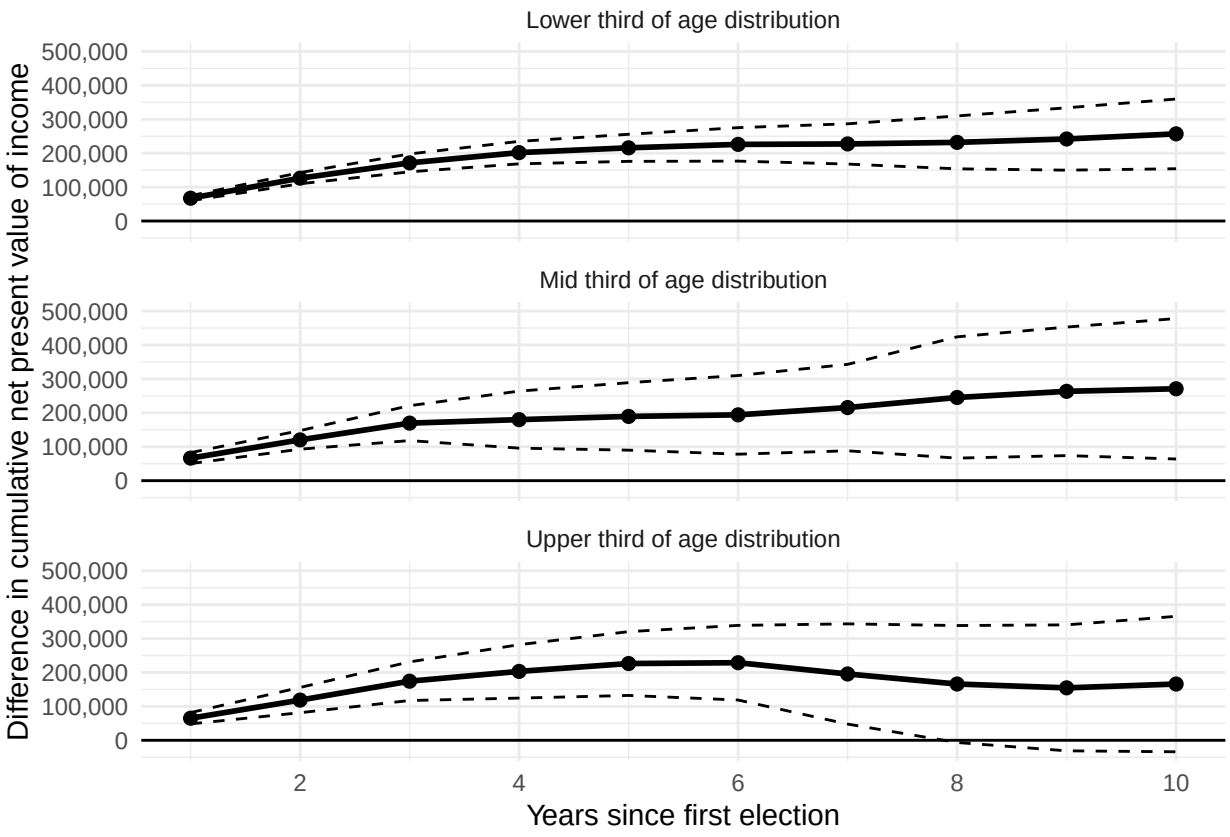


Figure J.4

```
#############################################################################
### PLOT BY AGE GROUP ###
### CALCULATE DIF IN MEANS ACROSS T AGE AGE THIRD
#############################################################################

## estimate effects within age groups
# calculate dif-in-means, create dfs of estimates and ad variables for labels etc
npv_124_cumu_age <- df_longrun %>%
  filter(t>0 & t<21) %>%
  group_by(t, age_third) %>%
  do(tidy(lm_robust(npv_124_cumu~elected, data=., clusters = cluster_id))) %>%
  filter(term=="elected")

npv_pens_124_cumu_age <- df_longrun %>%
  filter(t>0 & t<21) %>%
  group_by(t, age_third) %>%
  do(tidy(lm_robust(npv_pens_124_cumu~elected, data=., clusters = cluster_id))) %>%
  filter(term=="elected")

npv_258_cumu_age <- df_longrun %>%
```

```
  filter(t>0 & t<21) %>%
  group_by(t, age_third) %>%
  do(tidy(lm_robust(npv_258_cumu~elected, data=., clusters = cluster_id))) %>%
  filter(term=="elected")

npv_pens_258_cumu_age <- df_longrun %>%
  filter(t>0 & t<21) %>%
  group_by(t, age_third) %>%
  do(tidy(lm_robust(npv_pens_258_cumu~elected, data=., clusters = cluster_id))) %>%
  filter(term=="elected")

npv_124_cumu_age$rate <- "Interest rate of 1.24 percent"
npv_pens_124_cumu_age$rate <- "Interest rate of 1.24 percent"
npv_258_cumu_age$rate <- "Interest rate of 2.58 percent"
npv_pens_258_cumu_age$rate <- "Interest rate of 2.58 percent"

npv_124_cumu_age$outcome <- "Income excl. pension"
npv_258_cumu_age$outcome <- "Income excl. pension"
npv_pens_124_cumu_age$outcome <- "Income incl. pension"
npv_pens_258_cumu_age$outcome <- "Income incl. pension"


plot_df_age <- bind_rows(npv_124_cumu_age, npv_258_cumu_age,
                         npv_pens_124_cumu_age, npv_pens_258_cumu_age)


plot_df_age <- plot_df_age %>%
  dplyr::mutate(age_third = case_when(age_third==1~"Lower third of age distribution",
                                      age_third==2~"Mid third of age distribution",
                                      age_third==3~"Upper third of age distribution")) %>%
  dplyr::mutate(age_third = factor(age_third, levels = c("Lower third of age distribution",
                                                         "Mid third of age distribution",
                                                         "Upper third of age distribution"))) %>%
  dplyr::mutate(outcome = factor(outcome, levels = c("Income incl. pension", "Income excl. pension")))

################ MAIN AGE PLOT FOR NPV
################ 2.58 RATE INCLUDING PENSION / FOR MANUSCRIPT

plot_df_age %>%
  filter(t<11) %>%
  filter(rate=="Interest rate of 2.58 percent") %>%
  filter(outcome=="Income incl. pension") %>%
  ggplot(data=., aes(x = t, y = estimate, fill=outcome, scale = outcome)) +
  geom_point(size = 2, aes(shape = outcome)) +
  geom_line(size=1) +
  geom_line(aes(y = conf.low), linetype = 2) +
  geom_line(aes(y = conf.high), linetype = 2) +
  geom_hline(yintercept = 0, linetype=1) +
  theme_minimal() +
  theme(legend.position="none") +
  facet_wrap(~age_third, ncol=1) +
  scale_fill_grey(name="", start = 0.75, end = 0.35) +
  scale_color_grey(name="", start = 0.75, end = 0.35) +
```

```
  scale_shape_manual(name="", values =c(19,19,17,17)) +
  guides(alpha="none")  +
  scale_x_continuous(breaks = seq(2,20,2), labels = seq(2,20,2)) +
  coord_cartesian(ylim = c(NA,500000)) +
  scale_y_continuous(breaks = seq(-100000,500000,100000), label = scales::comma) +
  labs(x="Years since first election", y="Difference in cumulative net present value of income")
```