Replication - Main Tables

2025-06-15

Replication of the main tables

Table 1 - Incumbent 2010

Required libraries.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ---- tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ---------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(stargazer)
```

```
##
## Please cite as:
##
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
library(knitr)
library(broom)
library(haven)
library(fixest)
library(modelsummary)
library(gt)
library(webshot2)
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

Defining the control and dependent variables.

```
gpcontrols <- c("GP_population", "GP_lit", "GP_sc", "GP_st", "GP_nbvillages",
                "RES00_gender", "RES00_obc", "RES00_sc", "RES00_st",
                "RES10_obc", "RES10_sc", "RES10_st", "RES05_obc", "RES05_sc", "RES05_st")


incum_dep_vars1 <- c("INC05_running", "INC05_voteshare",
                     "INCSPOUSE05_running", "INCSPOUSE05_voteshare",
                     "INCOTHER05_running", "INCOTHER05_voteshare")
```

Loading the data and filtering it.

```
data <- read_dta("~/work/Electoral data cleaned.dta")


data_filtered <- data %>%
  filter(RES10_gender == 0 & SAMPLE_hhsurvey == 1 & GP_tag == 1 & INC05_can_run == 1) %>%
  mutate(
    FAMnotINC05_running = INCorFAM05_running - INC05_running,
    FAMnotINC05_voteshare = INCorFAM05_voteshare - INC05_voteshare,
    FAMnotINC05_won = INCorFAM05_won - INC05_won
  )
```

Function for the regression formulas.

```
create_formula <- function(dep_var, model_type) {
  base_controls <- paste(gpcontrols, collapse = " + ")

  if (model_type == "any_treatment") {
    formula_str <- paste(dep_var, "~ INT_treatment + RES05_gender + INT_treatment:RES05_gender +",
                         base_controls, "+ factor(district)")
  } else if (model_type == "gender_general") {
    formula_str <- paste(dep_var, "~ INT_treatment_gender + INT_treatment_general + RES05_gender + INT_
                         base_controls, "+ factor(district)")
  }

  return(as.formula(formula_str))
}
```

Function for the statistical tests.

```
calculate_tests <- function(model, model_type) {
  if (model_type == "any_treatment") {
    test1 <- tryCatch({
      car::linearHypothesis(model, "RES05_gender = 0")
    }, error = function(e) list(PrF = NA))
    pval1 <- if (!is.null(test1$PrF)) round(test1$`Pr(>F)`[2], 2) else NA

    test2 <- tryCatch({
      car::linearHypothesis(model, "INT_treatment:RES05_gender = 0")
    }, error = function(e) list(PrF = NA))
    pval2 <- if (!is.null(test2$PrF)) round(test2$`Pr(>F)`[2], 2) else NA

    test3 <- tryCatch({
      car::linearHypothesis(model, "INT_treatment = INT_treatment:RES05_gender")
    }, error = function(e) list(PrF = NA))
    pval3 <- if (!is.null(test3$PrF)) round(test3$`Pr(>F)`[2], 2) else NA

    return(list(pval1 = pval1, pval2 = pval2, pval3 = pval3))
  } else if (model_type == "gender_general") {
    test1 <- tryCatch({
      car::linearHypothesis(model, "INT_treatment_gender:RES05_gender = 0")
    }, error = function(e) list(PrF = NA))
    pval1 <- if (!is.null(test1$PrF)) round(test1$`Pr(>F)`[2], 2) else NA

    test2 <- tryCatch({
      car::linearHypothesis(model, "INT_treatment_general:RES05_gender = 0")
    }, error = function(e) list(PrF = NA))
    pval2 <- if (!is.null(test2$PrF)) round(test2$`Pr(>F)`[2], 2) else NA

    test3 <- tryCatch({
      car::linearHypothesis(model, "INT_treatment_gender = INT_treatment_general")
    }, error = function(e) list(PrF = NA))
    pval3 <- if (!is.null(test3$PrF)) round(test3$`Pr(>F)`[2], 2) else NA

    test4 <- tryCatch({
      car::linearHypothesis(model, "INT_treatment_gender:RES05_gender = INT_treatment_general:RES05_gen
    }, error = function(e) list(PrF = NA))
    pval4 <- if (!is.null(test4$PrF)) round(test4$`Pr(>F)`[2], 2) else NA

    return(list(pval1 = pval1, pval2 = pval2, pval3 = pval3, pval4 = pval4))
  }
}
```

Estimating the models, starting by initialising lists of results.

```
models_list <- list()
control_means <- list()
test_results <- list()
```

Models with “any treatment”:

```
for (i in 1:length(incum_dep_vars1)) {
  dep_var <- incum_dep_vars1[i]

  control_mean <- data_filtered %>%
    filter(INT_treatment == 0 & RES05_gender == 0) %>%
    summarise(mean = mean(!!sym(dep_var), na.rm = TRUE)) %>%
    pull(mean) %>%
    round(2)

  control_means[[i]] <- control_mean

  formula <- create_formula(dep_var, "any_treatment")
  model <- lm(formula, data = data_filtered)
  models_list[[i]] <- model

  test_results[[i]] <- calculate_tests(model, "any_treatment")
}
```

Models with “gender and general treatment”:

```r
for (i in 1:length(incum_dep_vars1)) {
  dep_var <- incum_dep_vars1[i]
  j <- i + length(incum_dep_vars1)

  control_means[[j]] <- control_means[[i]]

  formula <- create_formula(dep_var, "gender_general")
  model <- lm(formula, data = data_filtered)
  models_list[[j]] <- model

  test_results[[j]] <- calculate_tests(model, "gender_general")
}
```

Variables to display and selection of columns and summary statistics (means and test results).

```r
outregvar2 <- c("INT_treatment", "INT_treatment_gender", "INT_treatment_general", "RES05_gender", "INT_


col_names <- c("Incumbent Runs", "Incumbent Vote Share",
               "Incumbent Spouse Runs", "Incumbent Spouse Vote Share",
               "Other Family Member Runs", "Other Family Member Vote Share")


additional_lines <- list(
  c("Observations", sapply(models_list, function(x) nobs(x))),
  c("Mean in Control without GQ", unlist(control_means)),
  c("Treatment with GQ = Treat without GQ", sapply(test_results[1:length(incum_dep_vars1)], function(x)
  c("Gender Treat = General Treat without GQ", sapply(test_results[(length(incum_dep_vars1)+1):length(te
  c("Gender Treat = General Treat with GQ", sapply(test_results[(length(incum_dep_vars1)+1):length(test_
)
```

Generating the output table.

```r
stargazer(models_list,
          type = "text",
          column.labels = col_names,
          keep = outregvar2,
          add.lines = additional_lines,
          digits = 2,
          title = "Table 1: Effects on Incumbent and Family Candidate Entry",
          out = "Table1_Incumbent_2010_completed.txt")
```

```
##
## Table 1: Effects on Incumbent and Family Candidate Entry
## ====================================================================================================
##
##                                         ------------------------------------------------------------
##                                             INC05_running         INC05_voteshare      INCSPOUSE05_r
##                                             Incumbent Runs      Incumbent Vote Share  Incumbent Spous
##                                                  (1)                    (2)                    (3)
## ----------------------------------------------------------------------------------------------------
## INT_treatment                                  -0.26***               -6.27***                0.06
##                                                 (0.09)                 (2.35)                (0.07)
##
## INT_treatment_gender
```

```
##
##
## INT_treatment_general
##
##
## RES05_gender                                   -0.38***              -10.22***               0.36***
##                                                 (0.11)                 (2.65)                 (0.08)
##
## INT_treatment:RES05_gender                      0.36**                 9.13**                -0.27**
##                                                 (0.14)                 (3.52)                 (0.11)
##
## INT_treatment_gender:RES05_gender
##
##
## INT_treatment_general:RES05_gender
##
##
## ----------------------------------------------------------------------------------------------------
## Observations                                     152                      149                     152
## Mean in Control without GQ                       0.46                     10.1                   0.04
## Treatment with GQ = Treatment without GQ
## Gender Treat = General Treat without GQ
## Gender Treat = General Treat with GQ
## Observations                                     152                      149                     152
## R2                                               0.32                     0.39                   0.22
## Adjusted R2                                      0.21                     0.29                   0.10
## Residual Std. Error                        0.40 (df = 131)        9.94 (df = 128)        0.32 (df = 
## F Statistic                             3.02*** (df = 20; 131) 4.01*** (df = 20; 128) 1.80** (df = 2
## ====================================================================================================
## Note:
```

Narrower version of the table.

```r
# Table 1: Effects - Version Ultra-Compacte

# Table 1: Effects - Version Ultra-Compacte

# Table 1: Effects - Version Ultra-Compacte
# Noms de colonnes ultra-courts (2-3 caractères max)
col_names_ultra_short <- c("IR", "IV", "SR", "SV", "OR", "OV")

# Panel A: Average Effects
panel_A_models <- models_list[1:6]
panel_A_vars <- c("INT_treatment", "RES05_gender", "INT_treatment:RES05_gender")

# Panel A - Version ultra-compacte
stargazer(panel_A_models,
          type = "text",
          column.labels = col_names_ultra_short,  # Noms ultra-courts
          keep = panel_A_vars,
          digits = 2,
          title = "Panel A: Average Effects",
          covariate.labels = c("T", "F", "T×F"),  # T=Treatment, F=Female, T×F=Interaction
          omit.stat = c("ser", "adj.rsq", "f", "rsq", "n"),
          font.size = "tiny",
          no.space = TRUE,
          column.sep.width = "0pt",
          notes.append = FALSE,
          notes = "IR=Incumbent Run, IV=Incumbent Vote, SR=Spouse Run, SV=Spouse Vote, OR=Other Run, OV=
          star.cutoffs = c(0.05, 0.01, 0.001),  # Optionnel: ajuster les seuils de significativité
          header = FALSE)  # Supprime l'en-tête automatique de stargazer
```

```
##
## Panel A: Average Effects
## ====================================================================================================
##                                                       Dependent variable:
##       ----------------------------------------------------------------------------------------------
##       INC05_running INC05_voteshare INCSPOUSE05_running INCSPOUSE05_voteshare INCOTHER05_running INC
##            IR             IV                 SR                   SV                   OR
##            (1)            (2)                (3)                  (4)                 (5)
## ----------------------------------------------------------------------------------------------------
## T         -0.26**        -6.27**            0.06                 1.04                 0.11
##           (0.09)         (2.35)            (0.07)               (1.88)               (0.07)
## F        -0.38***       -10.22***          0.36***              5.76**               -0.15*
##           (0.11)         (2.65)            (0.08)               (2.12)               (0.07)
## T×F        0.36*          9.13*            -0.27*               -4.68                 0.03
##           (0.14)         (3.52)            (0.11)               (2.82)               (0.10)
## ====================================================================================================
## ====================================================================================================
## Note:                  IR=Incumbent Run, IV=Incumbent Vote, SR=Spouse Run, SV=Spouse Vote, OR=Other Ru
```

```
# Alternative encore plus compacte si nécessaire:
# Vous pouvez aussi utiliser des numéros:
col_names_numeric <- c("(1)", "(2)", "(3)", "(4)", "(5)", "(6)")

# Version avec numéros de colonnes
stargazer(panel_A_models,
          type = "text",
          column.labels = col_names_numeric,
          keep = panel_A_vars,
          digits = 2,
          title = "Panel A: Average Effects",
          covariate.labels = c("Treatment", "Female", "Treatment × Female"),
          omit.stat = c("ser", "adj.rsq", "f", "rsq", "n"),
          font.size = "tiny",
          no.space = TRUE,
          column.sep.width = "0pt",
          notes.append = FALSE,
          notes = "(1)-(2) Incumbent, (3)-(4) Spouse, (5)-(6) Other; Odd cols=Run, Even cols=Vote",
          header = FALSE)
```

```
##
## Panel A: Average Effects
## ====================================================================================================
##                                                                    Dependent variable:
##                    ---------------------------------------------------------------------------------
##                    INC05_running INC05_voteshare INCSPOUSE05_running INCSPOUSE05_voteshare INCOTHER05
##                         (1)            (2)                 (3)                  (4)                  (5
##                         (1)            (2)                 (3)                  (4)                  (5
## ----------------------------------------------------------------------------------------------------
## Treatment            -0.26***        -6.27***            0.06                 1.04                 0.1
##                       (0.09)          (2.35)            (0.07)               (1.88)                (0.
## Female               -0.38***        -10.22***          0.36***              5.76***             -0.15
##                       (0.11)          (2.65)            (0.08)               (2.12)                (0.
## Treatment × Female    0.36**          9.13**            -0.27**              -4.68*                 0.
##                       (0.14)          (3.52)            (0.11)               (2.82)                (0.
## ======================================================================================================
## ======================================================================================================
## Note:                                                (1)-(2) Incumbent, (3)-(4) Spouse, (5)-(6) Other
```

Table 2 - Performance 2010

```r
# Libraries
library(tidyverse)
library(fixest)
library(stargazer)
library(haven)
library(lmtest)
```

```
## Loading required package: zoo
```

```
## 
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
## 
##     as.Date, as.Date.numeric
```

```r
# ## DEFINITION OF THE MACROS ## #

# Control variables
gpcontrols <- c("GP_population", "GP_lit", "GP_sc", "GP_st", "GP_nbvillages",
                "RES00_gender", "RES00_obc", "RES00_sc", "RES00_st",
                "RES10_obc", "RES10_sc", "RES10_st", "RES05_obc", "RES05_sc", "RES05_st")

gpcontrols15 <- c(gpcontrols, "RES15_obc", "RES15_sc", "RES15_st")

# Regression variables
outregvar2 <- c("INT_treatment", "RES05_gender", "X_anytr_genderres05")
```

```r
# ## DATA PROCESSING ## #

# Loading the data. Change path accordingly to your workspace.
data <- read_dta("~/work/Electoral data cleaned.dta")

# Filtering the data
data_filtered <- data %>%
  filter(RES10_gender == 0, SAMPLE_hhsurvey == 1, GP_tag == 1, INC05_can_run == 1) %>%
  mutate(
    FAMnotINC05_running = INCFAM05_running - INC05_running,
    FAMnotINC05_voteshare = INCFAM05_voteshare - INC05_voteshare,
    FAMnotINC05_won = INCFAM05_won - INC05_won
  )

# Generate the PERFORMANCE INDICES of the program
# Explanation: used variables to build the indices are the data related to the success of implementatio
# Standardized measures about: participation, satisfied demand, waiting time, work done in the frame of
data_filtered <- data_filtered %>%
  mutate(
    # index_empl_svy_0 participation, unmet demand (men and women)
    index_empl_svy_0 = rowMeans(select(., std_HH_NREGA, std_HH_NREGA_unmet_demand_m, std_HH_NREGA_unmet_
    # index_empl_svy_1 unmet demand, waiting time, and work provided within the NREGA (men and women)
    index_empl_svy_1 = rowMeans(select(., std_HH_NREGA_unmet_demand, std_HH_NREGA_unmet_demand_m, std_H
    # index_empl_svy_2 work provided (men and women)
    index_empl_svy_2 = rowMeans(select(., std_HH_NREGA, std_HH_NREGA_work_m, std_HH_NREGA_work_f), na.r
    # index_empl_svy_3 unmet demand (men and women)
    index_empl_svy_3 = rowMeans(select(., std_HH_NREGA_unmet_demand_m, std_HH_NREGA_unmet_demand_f), na
  )

# Dependent variables
incum_dep_vars1 <- c("INC05_running", "INC05_voteshare", "INC05_won",
                     "INCSPOUSE05_running", "INCSPOUSE05_voteshare", "INCSPOUSE05_won",
                     "INCOTHER05_running", "INCOTHER05_voteshare", "INCOTHER05_won")

indices <- c("index_empl_svy_1")
```

```
# Starting lists to stock the upcoming results
models_list <- list()
control_means <- numeric(length(incum_dep_vars1) * length(indices))
pvals_1 <- numeric(length(incum_dep_vars1) * length(indices))
pvals_2 <- numeric(length(incum_dep_vars1) * length(indices))
effect_average <- numeric(length(incum_dep_vars1) * length(indices))
effect_good <- numeric(length(incum_dep_vars1) * length(indices))
effect_bad <- numeric(length(incum_dep_vars1) * length(indices))
```

```
# ## DOING THE REGRESSIONS ## #

i <- 0
for (x in 0:1) {
  for (dep_var in incum_dep_vars1) {
    for (index in indices) {
      i <- i + 1

      # control mean
      control_mean <- data_filtered %>%
        filter(INT_treatment == 0 & RES05_gender == x) %>%
        summarise(mean = mean(!!sym(dep_var), na.rm = TRUE)) %>%
        pull(mean) %>%
        round(2)

      control_means[i] <- control_mean

      # mean and standard error of the index
      index_stats <- data_filtered %>%
        filter(RES05_gender == x) %>%
        summarise(mean = mean(!!sym(index), na.rm = TRUE),
                  sd = sd(!!sym(index), na.rm = TRUE))

      index_mean <- round(index_stats$mean, 2)
```

```r
    index_sd <- round(index_stats$sd, 2)

    # interaction variables
    # explanation: interactions between performance at the time and gender of the incumbent in 2005 a
    # helps evaluating how the treatment effects vary depending on the gender and the performance of
    data_filtered <- data_filtered %>%
      mutate(
        TEMP_index = get(index),
        TEMP_X_res_index = RES05_gender * get(index),
        TEMP_X_anytr_index = INT_treatment * get(index),
        TEMP_X_anytr_res_index = INT_treatment * RES05_gender * get(index)
      )

    # checking that all the variables exist in the set
    all_vars <- c(dep_var, "INT_treatment", "TEMP_index", "TEMP_X_anytr_index", gpcontrols, "district
    if (all(all_vars %in% names(data_filtered))) {
      # model estimation
      formula <- as.formula(paste(dep_var, "~ INT_treatment + TEMP_index + TEMP_X_anytr_index +", pas
      model <- tryCatch({
        lm(formula, data = data_filtered %>% filter(RES05_gender == x))
      }, error = function(e) {
        message("Error in model fitting: ", e$message)
        NULL
      })

      if (!is.null(model)) {
        models_list[[i]] <- model

        # doing the tests
        test_1 <- tryCatch({
          waldtest(model, c("INT_treatment + TEMP_X_anytr_index" = 0, paste("TEMP_index", index_mean,
        }, error = function(e) {
          message("Error in test 1: ", e$message)
          NULL
        })

        if (!is.null(test_1)) {
          pvals_1[i] <- round(test_1$p.value, 2)
        } else {
          pvals_1[i] <- NA
        }

        test_2 <- tryCatch({
          waldtest(model, c("INT_treatment + TEMP_X_anytr_index" = 0))
        }, error = function(e) {
          message("Error in test 2: ", e$message)
          NULL
        })

        if (!is.null(test_2)) {
          pvals_2[i] <- round(test_2$p.value, 2)
        } else {
          pvals_2[i] <- NA
          }

          # effects
          effect_average[i] <- coef(model)["INT_treatment"] + coef(model)["TEMP_X_anytr_index"] * index_
          effect_good[i] <- coef(model)["INT_treatment"] + coef(model)["TEMP_X_anytr_index"] * (index_m
          effect_bad[i] <- coef(model)["INT_treatment"] + coef(model)["TEMP_X_anytr_index"] * (index_me

          # displaying said effects
          cat("Effects on outcome", dep_var, "\n")
          cat("Effect of treatment for average performing incumbent is", effect_average[i], "\n")
          cat("Effect of treatment for +1 sd performing incumbent is", effect_good[i], "\n")
          cat("Effect of treatment for -1 sd performing incumbent is", effect_bad[i], "\n")
        } else {
          message("Model fitting failed for ", dep_var)
        }
      } else {
        message("Some variables are missing in the dataset for ", dep_var)
      }
    }
  }
}
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "0", "TEMP_index
## = 0.01"
```

```
## Error in test 1: empty model specification
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): for
## numeric model specifications all values have to be >=1
```

```
## Error in test 2: empty model specification
```

```
## Effects on outcome INC05_running
## Effect of treatment for average performing incumbent is -0.2673244
## Effect of treatment for +1 sd performing incumbent is -0.1495474
## Effect of treatment for -1 sd performing incumbent is -0.3851014
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "0", "TEMP_index
## = 0.01"
```

```
## Error in test 1: empty model specification
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): for
## numeric model specifications all values have to be >=1
```

```
## Error in test 2: empty model specification
```

```
## Effects on outcome INC05_voteshare
## Effect of treatment for average performing incumbent is -6.748288
## Effect of treatment for +1 sd performing incumbent is -2.795807
## Effect of treatment for -1 sd performing incumbent is -10.70077
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "0", "TEMP_index
## = 0.01"
```

```
## Error in test 1: empty model specification
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): for
## numeric model specifications all values have to be >=1

## Error in test 2: empty model specification

## Effects on outcome INC05_won
## Effect of treatment for average performing incumbent is -0.01167305
## Effect of treatment for +1 sd performing incumbent is -0.002991184
## Effect of treatment for -1 sd performing incumbent is -0.02035492

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "0", "TEMP_index
## = 0.01"

## Error in test 1: empty model specification

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): for
## numeric model specifications all values have to be >=1

## Error in test 2: empty model specification

## Effects on outcome INCSPOUSE05_running
## Effect of treatment for average performing incumbent is 0.06395724
## Effect of treatment for +1 sd performing incumbent is 0.007988227
## Effect of treatment for -1 sd performing incumbent is 0.1199262

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "0", "TEMP_index
## = 0.01"

## Error in test 1: empty model specification

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): for
## numeric model specifications all values have to be >=1

## Error in test 2: empty model specification

## Effects on outcome INCSPOUSE05_voteshare
## Effect of treatment for average performing incumbent is 0.5693109
## Effect of treatment for +1 sd performing incumbent is -0.1275501
## Effect of treatment for -1 sd performing incumbent is 1.266172

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "0", "TEMP_index
## = 0.01"

## Error in test 1: empty model specification

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): for
## numeric model specifications all values have to be >=1

## Error in test 2: empty model specification

## Effects on outcome INCSPOUSE05_won
## Effect of treatment for average performing incumbent is 0.0003176332
## Effect of treatment for +1 sd performing incumbent is 0.004545843
## Effect of treatment for -1 sd performing incumbent is -0.003910577

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "0", "TEMP_index
## = 0.01"

## Error in test 1: empty model specification
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): for
## numeric model specifications all values have to be >=1

## Error in test 2: empty model specification

## Effects on outcome INCOTHER05_running
## Effect of treatment for average performing incumbent is 0.1038207
## Effect of treatment for +1 sd performing incumbent is 0.07884448
## Effect of treatment for -1 sd performing incumbent is 0.128797

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "0", "TEMP_index
## = 0.01"

## Error in test 1: empty model specification

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): for
## numeric model specifications all values have to be >=1

## Error in test 2: empty model specification

## Effects on outcome INCOTHER05_voteshare
## Effect of treatment for average performing incumbent is 1.41262
## Effect of treatment for +1 sd performing incumbent is 1.690713
## Effect of treatment for -1 sd performing incumbent is 1.134528

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "0", "TEMP_index
## = 0.01"

## Error in test 1: empty model specification

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): for
## numeric model specifications all values have to be >=1

## Error in test 2: empty model specification

## Effects on outcome INCOTHER05_won
## Effect of treatment for average performing incumbent is 0.06241062
## Effect of treatment for +1 sd performing incumbent is 0.09167475
## Effect of treatment for -1 sd performing incumbent is 0.03314649

## Error in test 1: there are aliased coefficients in the model

## Error in test 2: there are aliased coefficients in the model

## Effects on outcome INC05_running
## Effect of treatment for average performing incumbent is 0.08562395
## Effect of treatment for +1 sd performing incumbent is 0.01559434
## Effect of treatment for -1 sd performing incumbent is 0.1556536

## Error in test 1: there are aliased coefficients in the model
## Error in test 2: there are aliased coefficients in the model

## Effects on outcome INC05_voteshare
## Effect of treatment for average performing incumbent is 1.637864
## Effect of treatment for +1 sd performing incumbent is 0.788168
## Effect of treatment for -1 sd performing incumbent is 2.487559

## Error in test 1: there are aliased coefficients in the model
## Error in test 2: there are aliased coefficients in the model
## Effects on outcome INC05_won
## Effect of treatment for average performing incumbent is -0.01505433
## Effect of treatment for +1 sd performing incumbent is -0.002202576
## Effect of treatment for -1 sd performing incumbent is -0.02790608
```

```
## Error in test 1: there are aliased coefficients in the model
## Error in test 2: there are aliased coefficients in the model
```

```
## Effects on outcome INCSPOUSE05_running
## Effect of treatment for average performing incumbent is -0.2432261
## Effect of treatment for +1 sd performing incumbent is -0.1874189
## Effect of treatment for -1 sd performing incumbent is -0.2990334
```

```
## Error in test 1: there are aliased coefficients in the model
## Error in test 2: there are aliased coefficients in the model
```

```
## Effects on outcome INCSPOUSE05_voteshare
## Effect of treatment for average performing incumbent is -5.396206
## Effect of treatment for +1 sd performing incumbent is -7.982221
## Effect of treatment for -1 sd performing incumbent is -2.810192
```

```
## Error in test 1: there are aliased coefficients in the model
## Error in test 2: there are aliased coefficients in the model
```

```
## Effects on outcome INCSPOUSE05_won
## Effect of treatment for average performing incumbent is 0.0255038
## Effect of treatment for +1 sd performing incumbent is 0.04624032
## Effect of treatment for -1 sd performing incumbent is 0.004767284
```

```
## Error in test 1: there are aliased coefficients in the model
## Error in test 2: there are aliased coefficients in the model
```

```
## Effects on outcome INCOTHER05_running
## Effect of treatment for average performing incumbent is 0.1294562
## Effect of treatment for +1 sd performing incumbent is 0.271624
## Effect of treatment for -1 sd performing incumbent is -0.01271158
```

```
## Error in test 1: there are aliased coefficients in the model
## Error in test 2: there are aliased coefficients in the model
```

```
## Effects on outcome INCOTHER05_voteshare
## Effect of treatment for average performing incumbent is 2.548712
## Effect of treatment for +1 sd performing incumbent is 4.972919
## Effect of treatment for -1 sd performing incumbent is 0.1245048
```

```
## Error in test 1: there are aliased coefficients in the model
## Error in test 2: there are aliased coefficients in the model
```

```
## Effects on outcome INCOTHER05_won
## Effect of treatment for average performing incumbent is 0.02115388
## Effect of treatment for +1 sd performing incumbent is 0.02040723
## Effect of treatment for -1 sd performing incumbent is 0.02190053
```

```r
# ## GENERATING THE OUTPUT TABLE ## #

stargazer(models_list,
          type = "text",
          column.labels = paste("Model", 1:length(models_list)),
          keep = c("INT_treatment", "TEMP_index", "TEMP_X_anytr_index"),
          add.lines = list(
            c("District FE", rep("Yes", length(models_list))),
            c("GP Controls", rep("Yes", length(models_list))),
            c("Mean in Control not WR in 2005", control_means),
            c("Test Treat Effect", pvals_1),
            c("Test Perf Effect in Treat", pvals_2)
          ),
          digits = 2,
          title = "Table 2: Performance - 2010",
          # change output path accordingly to your workspace
          out = file.path("~/work/Rajasthan-Voters-Replication/Table2_Performance_2010.txt"))
```

```
##
## Table 2: Performance - 2010
## ====================================================================================================
##
##                                ----------------------------------------------------------------------
##                                    INC05_running        INC05_voteshare        INC05_won      INCSPOU
##                                       Model 1               Model 2             Model 3
##                                         (1)                   (2)                 (3)
## ----------------------------------------------------------------------------------------------------
## INT_treatment                         -0.27**               -6.83**              -0.01
##                                       (0.10)                (2.58)               (0.06)
##
## TEMP_index                             -0.09                 -4.05               -0.09
##                                       (0.13)                (3.30)               (0.07)
##
## TEMP_X_anytr_index                     0.24                  8.07                 0.02
##                                       (0.20)                (4.96)               (0.11)
##
## ----------------------------------------------------------------------------------------------------
## District FE                             Yes                   Yes                 Yes
## GP Controls                             Yes                   Yes                 Yes
## Mean in Control not WR in 2005         0.46                  10.1                 0.06
## Test Treat Effect
## Test Perf Effect in Treat
## Observations                            92                    90                   92
## R2                                     0.40                  0.51                 0.22
## Adjusted R2                            0.23                  0.36                -0.004
## Residual Std. Error                0.42 (df = 71)        10.51 (df = 69)      0.23 (df = 71)     0.22
## F Statistic                     2.36*** (df = 20; 71) 3.52*** (df = 20; 69) 0.98 (df = 20; 71) 1.17 (
## ====================================================================================================
## Note:
```

Table 3 - Challengers 2010

Table 4- Candidates 2015

```r
# Packages to install if necessary
install.packages(c("tidyverse", "haven", "fixest","stargazer"))
```

```
## Installing packages into '/usr/local/lib/R/site-library'
## (as 'lib' is unspecified)
```

```
#Libraries
library(tidyverse)
library(fixest)
library(stargazer)
library(haven)



## MACROS

# Controls
gpcontrols <- c("GP_population", "GP_lit", "GP_sc", "GP_st", "GP_nbvillages",
                "RES00_gender", "RES00_obc", "RES00_sc", "RES00_st",
                "RES10_obc", "RES10_sc", "RES10_st", "RES05_obc", "RES05_sc", "RES05_st")

gpcontrols15 <- c(gpcontrols, "RES15_obc", "RES15_sc", "RES15_st")

# Regression variables
outregvar2 <- c("INT_treatment", "RES05_gender", "X_anytr_genderres05")


## DATA PROCESSING

# Loading the data. Change path depending on your workspace.
data <- read_dta("~/work/Electoral data 2015 cleaned.dta")

# Filtering the data
data_filtered <- data %>%
  filter(RES10_gender == 0, GP_tag == 1, RES15_gender == 0) %>%
  mutate(
    INC10_can_run = 1,
    INC10_can_run = ifelse(ELEC10_won_female == 0 & RES15_gender == 1, 0, INC10_can_run),
    INC10_can_run = ifelse(ELEC10_won_sc == 0 & RES15_sc == 1, 0, INC10_can_run),
    INC10_can_run = ifelse(ELEC10_won_st == 0 & RES15_st == 1, 0, INC10_can_run)
  )

# Generate new variables
for (var in c("INT_treatment", "X_anytr_genderres05", "RES05_gender")) {
  data_filtered <- data_filtered %>%
    mutate(!!paste0("X15_", var) := get(var) * (RES15_gender == 1))
}

outregvar15 <- c("INT_treatment", "RES05_gender", "X_anytr_genderres05", "RES15_gender",
                 "X15_INT_treatment", "X15_RES05_gender", "X15_X_anytr_genderres05")

# Dependent variables
dep_vars <- c("ELEC15_nbcands", "ELEC15_incum10_running", "ELEC15_voteshare_incum10",
              "ELEC15_prop_cand2010", "ELEC15_voteshare_cand2010", "ELEC15_prop_female",
              "ELEC15_voteshare_female", "ELEC15_prop_nongen", "ELEC15_voteshare_nongen")
```

```
# List to stock the results:
models_list <- list()
control_means <- numeric(length(dep_vars))
pvals <- numeric(length(dep_vars))
```

```
## DOING THE REGRESSIONS


for (i in seq_along(dep_vars)) {
  dep_var <- dep_vars[i]

  # control mean
  control_mean <- data_filtered %>%
    filter(INT_treatment == 0 & RES05_gender == 0) %>%
    summarise(mean = mean(!!sym(dep_var), na.rm = TRUE)) %>%
    pull(mean) %>%
    round(2)

  control_means[i] <- control_mean

  # model estimation
  formula <- as.formula(paste(dep_var, "~", paste(c(outregvar2, gpcontrols15), collapse = " + "), "+ fac
  model <- lm(formula, data = data_filtered)
  models_list[[i]] <- model

  # do the test
  test_result <- summary(lm(test = RES05_gender + X_anytr_genderres05, data = model$model))$coefficients
  pvals[i] <- round(test_result, 2)
}
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'test' will be disregarded
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'test' will be disregarded
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'test' will be disregarded
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'test' will be disregarded
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'test' will be disregarded
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'test' will be disregarded
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'test' will be disregarded
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'test' will be disregarded
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'test' will be disregarded
```

```
stargazer(models_list,
          type = "text",
          column.labels = paste("Model", 1:length(dep_vars)),
          keep = outregvar2,
          add.lines = list(
            c("District FE", rep("Yes", length(dep_vars))),
            c("GP Controls", rep("Yes", length(dep_vars))),
            c("Mean in Control not WR in 2015", control_means),
            c("Test Treat Effect in WR=Treat Effect in NWR", pvals)
          ),
          digits = 2,
          title = "Table 4: Effects on Candidates - 2015",
          out = file.path("~/work/Rajasthan-Voters-Replication/Table4_Candidates_2015.txt"))
```

```
##
## Table 4: Effects on Candidates - 2015
## ====================================================================================================
##
##                                              --------------------------------------------------------
##                                              ELEC15_nbcands ELEC15_incum10_running ELEC15_voteshare_i
##                                                 Model 1            Model 2                 Model 3
##                                                   (1)                (2)                     (3)
## ----------------------------------------------------------------------------------------------------
## INT_treatment                                      0.35               0.07                   2.62**
##                                                   (1.21)             (0.05)                  (1.29)
##
## RES05_gender                                       0.82             0.12***                  2.42**
##                                                   (1.13)             (0.05)                  (1.20)
##
## X_anytr_genderres05                               -2.57              -0.14*                  -3.72*
##                                                   (1.89)             (0.08)                  (2.00)
##
## ----------------------------------------------------------------------------------------------------
## District FE                                        Yes                Yes                     Yes
## GP Controls                                        Yes                Yes                     Yes
## Mean in Control not WR in 2015                     7.83                0                       0
## Test Treat Effect in WR=Treat Effect in NWR      0.77               0.16                    0.05
## Observations                                       89                 89                      89
## R2                                                 0.32               0.29                    0.27
## Adjusted R2                                        0.08               0.03                    0.02
## Residual Std. Error (df = 65)                      3.65               0.15                    3.87
## F Statistic (df = 23; 65)                          1.32               1.13                    1.06
## ====================================================================================================
## Note:
```

Table 5 - Voters perception