Replication - Main Tables

2025-06-15

Replication of the main tables

Table 1 - Incumbent 2010

Required libraries.

```r
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
```

```r
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
```

```r
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.

```r
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.

```r
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.

```r
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.

```r
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_t


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_ru
##                                             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 = Treat 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 = 1
## F Statistic                          3.02*** (df = 20; 131) 4.01*** (df = 20; 128) 1.80** (df = 20
## ====================================================================================================
## Note:
```

Non formatted version of the table, to fit into the PDF.

```
## CREATE OUTPUT TABLE FOR INCUMBENT ANALYSIS
create_outreg_table <- function(models_list, incum_dep_vars1, outregvar2, control_means, test_results) {

  # Function to extract model results
  extract_model_results <- function(model) {
    if (is.null(model)) return(NULL)

    summary_model <- summary(model)
    coef_table <- summary_model$coefficients

    results <- list()
    for (var in outregvar2) {
      matching_vars <- rownames(coef_table)[grepl(paste0("^", gsub(":", ":", var), "$"), rownames(coef_
      if (length(matching_vars) > 0) {
        var_name <- matching_vars[1]
        coef_val <- coef_table[var_name, "Estimate"]
        se_val <- coef_table[var_name, "Std. Error"]
        pval <- coef_table[var_name, "Pr(>|t|)"]

        # Adding significance stars
        stars <- if (pval < 0.01) "***" else if (pval < 0.05) "**" else if (pval < 0.1) "*" else ""
```

```r
      results[[var]] <- list(
        coef = round(coef_val, 3),
        se = round(se_val, 3),
        pval = pval,
        stars = stars,
        coef_formatted = paste0(round(coef_val, 3), stars),
        se_formatted = paste0("(", round(se_val, 3), ")")
      )
    } else {
      results[[var]] <- list(
        coef = NA,
        se = NA,
        pval = NA,
        stars = "",
        coef_formatted = "NA",
        se_formatted = "(NA)"
      )
    }
  }
  return(results)
}

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

# Initialize the final table
final_table <- data.frame(
  Variable = character(),
  stringsAsFactors = FALSE
)

# Add columns for each dependent variable (both any_treatment and gender_general models)
for (i in 1:length(incum_dep_vars1)) {
  final_table[[paste0(col_names[i], "_Any")]] <- character()
  final_table[[paste0(col_names[i], "_Detailed")]] <- character()
}

# Extract results for each variable
for (var in outregvar2) {
  # Coefficient row
  coef_row <- data.frame(Variable = var, stringsAsFactors = FALSE)
  se_row <- data.frame(Variable = paste0("  ", var, "_se"), stringsAsFactors = FALSE)

  # For each dependent variable
  for (i in 1:length(incum_dep_vars1)) {
    # Any treatment model (first 6 models)
    any_results <- extract_model_results(models_list[[i]])
    coef_row[[paste0(col_names[i], "_Any")]] <- if (!is.null(any_results[[var]])) any_results[[var]]$c
    se_row[[paste0(col_names[i], "_Any")]] <- if (!is.null(any_results[[var]])) any_results[[var]]$se_

    # Gender/General treatment model (models 7-12)
```

```
      detailed_results <- extract_model_results(models_list[[i + length(incum_dep_vars1)]])
      coef_row[[paste0(col_names[i], "_Detailed")]] <- if (!is.null(detailed_results[[var]])) detailed_
      se_row[[paste0(col_names[i], "_Detailed")]] <- if (!is.null(detailed_results[[var]])) detailed_res
    }

    final_table <- rbind(final_table, coef_row, se_row)
  }

  # Add additional statistics rows
  # Observations
  obs_row <- data.frame(Variable = "Observations", stringsAsFactors = FALSE)
  for (i in 1:length(incum_dep_vars1)) {
    obs_row[[paste0(col_names[i], "_Any")]] <- nobs(models_list[[i]])
    obs_row[[paste0(col_names[i], "_Detailed")]] <- nobs(models_list[[i + length(incum_dep_vars1)]])
  }
  final_table <- rbind(final_table, obs_row)

  # Control means
  mean_row <- data.frame(Variable = "Mean in Control without GQ", stringsAsFactors = FALSE)
  for (i in 1:length(incum_dep_vars1)) {
    mean_row[[paste0(col_names[i], "_Any")]] <- control_means[[i]]
    mean_row[[paste0(col_names[i], "_Detailed")]] <- control_means[[i]]
  }
  final_table <- rbind(final_table, mean_row)

  # Test results
  test1_row <- data.frame(Variable = "Treatment with GQ = Treat without GQ", stringsAsFactors = FALSE)
  for (i in 1:length(incum_dep_vars1)) {
    test1_row[[paste0(col_names[i], "_Any")]] <- test_results[[i]]$pval3
    test1_row[[paste0(col_names[i], "_Detailed")]] <- ""
  }
  final_table <- rbind(final_table, test1_row)

  test2_row <- data.frame(Variable = "Gender Treat = General Treat without GQ", stringsAsFactors = FALS
  for (i in 1:length(incum_dep_vars1)) {
    test2_row[[paste0(col_names[i], "_Any")]] <- ""
    test2_row[[paste0(col_names[i], "_Detailed")]] <- test_results[[i + length(incum_dep_vars1)]]$pval3
  }
  final_table <- rbind(final_table, test2_row)

  test3_row <- data.frame(Variable = "Gender Treat = General Treat with GQ", stringsAsFactors = FALSE)
  for (i in 1:length(incum_dep_vars1)) {
    test3_row[[paste0(col_names[i], "_Any")]] <- ""
    test3_row[[paste0(col_names[i], "_Detailed")]] <- test_results[[i + length(incum_dep_vars1)]]$pval4
  }
  final_table <- rbind(final_table, test3_row)

  return(final_table)
}

# Create the main table
main_table <- create_outreg_table(models_list, incum_dep_vars1, outregvar2, control_means, test_results
```

```r
# Display the table
print(main_table)
```

```
##                                    Variable
## 1                            INT_treatment
## 2                         INT_treatment_se
## 3                     INT_treatment_gender
## 4                  INT_treatment_gender_se
## 5                    INT_treatment_general
## 6                 INT_treatment_general_se
## 7                             RES05_gender
## 8                          RES05_gender_se
## 9               INT_treatment:RES05_gender
## 10           INT_treatment:RES05_gender_se
## 11       INT_treatment_gender:RES05_gender
## 12    INT_treatment_gender:RES05_gender_se
## 13      INT_treatment_general:RES05_gender
## 14   INT_treatment_general:RES05_gender_se
## 15                            Observations
## 16              Mean in Control without GQ
## 17    Treatment with GQ = Treat without GQ
## 18 Gender Treat = General Treat without GQ
## 19    Gender Treat = General Treat with GQ
##    Incumbent Runs_Any Incumbent Runs_Detailed
## 1           -0.262***                      NA
## 2             (0.094)                    (NA)
## 3                  NA                 -0.242*
## 4                (NA)                 (0.123)
## 5                  NA                -0.278**
## 6                (NA)                 (0.111)
## 7           -0.376***               -0.381***
## 8             (0.107)                 (0.108)
## 9             0.357**                      NA
## 10            (0.141)                    (NA)
## 11                 NA                  0.38**
## 12               (NA)                 (0.172)
## 13                 NA                  0.309*
## 14               (NA)                 (0.177)
## 15                152                     152
## 16               0.46                    0.46
## 17               <NA>
## 18                                       <NA>
## 19                                       <NA>
##    Incumbent Vote Share_Any Incumbent Vote Share_Detailed
## 1                 -6.265***                            NA
## 2                   (2.353)                          (NA)
## 3                        NA                        -4.275
## 4                      (NA)                        (3.04)
## 5                        NA                      -7.81***
## 6                      (NA)                       (2.787)
## 7                -10.223***                    -10.531***
## 8                   (2.652)                       (2.659)
## 9                   9.127**                            NA
## 10                  (3.522)                          (NA)
## 11                       NA                       8.925**
## 12                     (NA)                        (4.24)
## 13                       NA                        8.055*
## 14                     (NA)                       (4.461)
## 15                      149                           149
## 16                     10.1                          10.1
## 17                     <NA>
## 18                                                   <NA>
## 19                                                   <NA>
##    Incumbent Spouse Runs_Any Incumbent Spouse Runs_Detailed
## 1                      0.057                             NA
## 2                    (0.074)                           (NA)
## 3                         NA                          0.006
## 4                       (NA)                        (0.097)
## 5                         NA                          0.094
## 6                       (NA)                        (0.087)
## 7                   0.364***                       0.367***
## 8                    (0.084)                        (0.084)
## 9                   -0.274**                             NA
## 10                   (0.111)                           (NA)
## 11                        NA                         -0.176
## 12                      (NA)                        (0.135)
## 13                        NA                      -0.382***
## 14                      (NA)                        (0.138)
## 15                       152                            152
## 16                      0.04                           0.04
## 17                      <NA>
## 18                                                     <NA>
## 19                                                     <NA>
##    Incumbent Spouse Vote Share_Any
## 1                            1.044
## 2                          (1.884)
## 3                               NA
## 4                             (NA)
## 5                               NA
## 6                             (NA)
## 7                         5.761***
## 8                          (2.124)
## 9                          -4.683*
## 10                          (2.82)
## 11                              NA
## 12                            (NA)
## 13                              NA
## 14                            (NA)
## 15                             149
## 16                            0.73
## 17                            <NA>
## 18
## 19
##    Incumbent Spouse Vote Share_Detailed
## 1                                    NA
## 2                                  (NA)
## 3                                 0.695
## 4                               (2.429)
## 5                                 1.315
## 6                               (2.226)
## 7                              5.707***
## 8                               (2.124)
## 9                                    NA
## 10                                 (NA)
## 11                               -2.318
## 12                              (3.387)
## 13                             -8.078**
## 14                              (3.563)
## 15                                  149
## 16                                 0.73
## 17
## 18                                 <NA>
## 19                                 <NA>
##    Other Family Member Runs_Any
## 1                        0.112*
## 2                       (0.066)
## 3                            NA
## 4                          (NA)
## 5                            NA
## 6                          (NA)
## 7                      -0.148**
## 8                       (0.075)
## 9                          0.03
## 10                      (0.099)
## 11                           NA
## 12                         (NA)
## 13                           NA
## 14                         (NA)
## 15                          152
## 16                         0.08
## 17                         <NA>
## 18
## 19
##    Other Family Member Runs_Detailed
## 1                                 NA
## 2                               (NA)
## 3                              0.109
## 4                            (0.087)
## 5                              0.114
## 6                            (0.078)
## 7                            -0.149*
## 8                            (0.076)
## 9                                 NA
## 10                              (NA)
## 11                             0.046
## 12                           (0.121)
## 13                             0.009
## 14                           (0.124)
## 15                               152
## 16                              0.08
## 17
## 18                              <NA>
## 19                              <NA>
##    Other Family Member Vote Share_Any
## 1                               1.826
## 2                             (1.505)
## 3                                  NA
## 4                                (NA)
## 5                                  NA
## 6                                (NA)
## 7                              -2.047
## 8                             (1.697)
## 9                               1.777
## 10                            (2.254)
## 11                                 NA
## 12                               (NA)
## 13                                 NA
## 14                               (NA)
## 15                                149
## 16                               0.86
## 17                               <NA>
## 18
## 19
##    Other Family Member Vote Share_Detailed
## 1                                       NA
## 2                                     (NA)
## 3                                    1.277
## 4                                  (1.931)
## 5                                    2.254
## 6                                   (1.77)
## 7                                   -2.072
## 8                                  (1.689)
## 9                                       NA
## 10                                    (NA)
## 11                                   4.222
## 12                                 (2.693)
## 13                                  -1.602
## 14                                 (2.833)
## 15                                     149
## 16                                    0.86
## 17
## 18                                    <NA>
## 19                                    <NA>
```

Table 2 - Performance 2010

Required 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
```

Defining the macros: control variables and variables related to the regression.

```
# 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")

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

Data processing: uploading, filtering.

```
# 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
  )
```

Generation of the performance indices of the program.

```
data_filtered <- data_filtered %>%
  mutate(
    index_empl_svy_0 = rowMeans(select(., std_HH_NREGA, std_HH_NREGA_unmet_demand_m, std_HH_NREGA_unmet_
    index_empl_svy_1 = rowMeans(select(., std_HH_NREGA_unmet_demand, std_HH_NREGA_unmet_demand_m, std_HH
    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 = 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_0", "index_empl_svy_1", "index_empl_svy_2", "index_empl_svy_3")
```

Initialization of the lists for 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.

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

      # interaction variables
      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_mea
          }, 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_me
          effect_bad[i] <- coef(model)["INT_treatment"] + coef(model)["TEMP_X_anytr_index"] * (index_mea

          # 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: "INT_treatment +
## TEMP_X_anytr_index = 0", "TEMP_index = -0.01"

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "INT_treatment +
## TEMP_X_anytr_index = 0"

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "INT_treatment +
## TEMP_X_anytr_index = 0"
## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "INT_treatment +
## TEMP_X_anytr_index = 0"

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]):
## terms specified that are not in the model: "INT_treatment +
## TEMP_X_anytr_index = 0"
## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

## Error in test 1: empty model specification

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

## Error in test 2: empty model specification

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Generation of a table displaying the coefficients of interest.

```r
# Selection of the right models to display
panel_A_models <- models_list[c(1, 5, 13, 17, 25, 29)]
panel_B_models <- models_list[c(37, 41, 49, 53, 61, 65)]

# Panel A
panel_A <- stargazer(
  panel_A_models,
  type = "text",
  column.labels = c("Model 1", "Model 5", "Model 13", "Model 17", "Model 25", "Model 29"),
  keep = c("INT_treatment", "TEMP_index", "TEMP_X_anytr_index"),
  add.lines = list(
    c("District FE", rep("Yes", length(panel_A_models))),
    c("GP Controls", rep("Yes", length(panel_A_models))),
    c("Mean in Control not WR in 2005", control_means[c(1, 5, 13, 17, 25, 29)]),
    c("Test Treat Effect", pvals_1[c(1, 5, 13, 17, 25, 29)]),
    c("Test Perf Effect in Treat", pvals_2[c(1, 5, 13, 17, 25, 29)])
  ),
  digits = 2,
  title = "Panel A: GP without Gender Quota in 2005",
  single.row = TRUE
)
```

```
##
## Panel A: GP without Gender Quota in 2005
## =====================================================================================================
##                                                                                     Dependent variabl
##                                ----------------------------------------------------------------------
##                                    INC05_running        INC05_voteshare    INCSPOUSE05_running INCSP
##                                       Model 1               Model 5             Model 13
##                                         (1)                   (2)                  (3)
## -----------------------------------------------------------------------------------------------------
## INT_treatment                     -0.27** (0.10)        -6.76*** (2.52)        0.07 (0.05)          0
## TEMP_index                         -0.09 (0.07)          -3.03* (1.71)        0.01 (0.04)         -0
## TEMP_X_anytr_index                 0.24** (0.11)         7.05** (2.82)       -0.06 (0.06)         -0
## -----------------------------------------------------------------------------------------------------
## District FE                              Yes                   Yes                  Yes
## GP Controls                              Yes                   Yes                  Yes
## Mean in Control not WR in 2005          0.46                  10.1                 0.04
## Test Treat Effect
## Test Perf Effect in Treat
## Observations                             92                    90                   92
## R2                                      0.42                  0.53                 0.24
## Adjusted R2                             0.26                  0.39                 0.03
## Residual Std. Error               0.42 (df = 71)        10.25 (df = 69)      0.22 (df = 71)       4.
## F Statistic                    2.59*** (df = 20; 71) 3.88*** (df = 20; 69) 1.13 (df = 20; 71)   0.63
## =====================================================================================================
## Note:
```

```
# Panel B
panel_B <- stargazer(
  panel_B_models,
  type = "text",
  column.labels = c("Model 37", "Model 41", "Model 49", "Model 53", "Model 61", "Model 65"),
  keep = c("INT_treatment", "TEMP_index", "TEMP_X_anytr_index"),
  add.lines = list(
    c("District FE", rep("Yes", length(panel_B_models))),
    c("GP Controls", rep("Yes", length(panel_B_models))),
    c("Mean in Control not WR in 2005", control_means[c(37, 41, 49, 53, 61, 65)]),
    c("Test Treat Effect", pvals_1[c(37, 41, 49, 53, 61, 65)]),
    c("Test Perf Effect in Treat", pvals_2[c(37, 41, 49, 53, 61, 65)])
  ),
  digits = 2,
  title = "Panel B: GP with Gender Quota in 2005",
  single.row = TRUE
)
```

```
##
## Panel B: GP with Gender Quota in 2005
## =====================================================================================================
##                                                                                  Dependent variable:
##                                ----------------------------------------------------------------------
##                                  INC05_running     INC05_voteshare   INCSPOUSE05_running INCSPOUSE05_
##                                     Model 37           Model 41           Model 49             Model
##                                       (1)                (2)                 (3)                  (4
## -----------------------------------------------------------------------------------------------------
## INT_treatment                     0.15 (0.11)        2.48 (2.53)       -0.27** (0.13)        -5.82 (
## TEMP_index                        -0.03 (0.12)       -0.18 (2.71)        0.07 (0.14)          4.57 (
## TEMP_X_anytr_index                -0.13 (0.14)       -2.54 (3.23)       -0.09 (0.17)         -5.11 (
## -----------------------------------------------------------------------------------------------------
## District FE                           Yes                Yes                 Yes                  Ye
## GP Controls                           Yes                Yes                 Yes                  Ye
## Mean in Control not WR in 2005        0.15               2.5                0.33                 6.1
## Test Treat Effect
## Test Perf Effect in Treat
## Observations                           60                 59                 60                   59
## R2                                    0.37               0.39               0.34                 0.2
## Adjusted R2                           0.07               0.09               0.03                 -0.
## Residual Std. Error              0.36 (df = 40)     8.13 (df = 39)     0.42 (df = 40)       11.47 (d
## F Statistic                    1.23 (df = 19; 40) 1.31 (df = 19; 39) 1.09 (df = 19; 40)   0.84 (df =
## ====================================================================================================
## Note:
```

```r
# Combine the two in one txt doc
combined_output <- c(panel_A, "\n\n", panel_B)
writeLines(combined_output, file.path("~/work/Rajasthan-Voters-Replication/Table2_Performance_2010_compl
```

Non formatted table, to fit into the PDF.

```r
create_reduced_performance_table <- function(models_list, control_means, pvals_1, pvals_2) {

  # Function to extract model results
  extract_model_results <- function(model) {
    if (is.null(model)) return(NULL)

    summary_model <- summary(model)
    coef_table <- summary_model$coefficients

    target_vars <- c("INT_treatment", "TEMP_index", "TEMP_X_anytr_index")

    results <- list()
    for (var in target_vars) {
      if (var %in% rownames(coef_table)) {
        coef_val <- coef_table[var, "Estimate"]
        se_val <- coef_table[var, "Std. Error"]
        pval <- coef_table[var, "Pr(>|t|)"]

        stars <- if (pval < 0.01) "***" else if (pval < 0.05) "**" else if (pval < 0.1) "*" else ""

        results[[var]] <- list(
          coef_formatted = paste0(round(coef_val, 3), stars),
          se_formatted = paste0("(", round(se_val, 3), ")")
        )
      } else {
        results[[var]] <- list(
          coef_formatted = "NA",
          se_formatted = "(NA)"
        )
      }
    }
    return(results)
  }

  # Selection of models to display
  panel_A_indices <- c(1, 5, 13, 17, 25, 29)
```

```r
panel_B_indices <- c(37, 41, 49, 53, 61, 65)

# Column names for the selected models
panel_A_cols <- c("Model_1", "Model_5", "Model_13", "Model_17", "Model_25", "Model_29")
panel_B_cols <- c("Model_37", "Model_41", "Model_49", "Model_53", "Model_61", "Model_65")

# Create Panel A table
panel_A_table <- data.frame(Variable = character(), stringsAsFactors = FALSE)
for (col in panel_A_cols) {
  panel_A_table[[col]] <- character()
}

# Create Panel B table
panel_B_table <- data.frame(Variable = character(), stringsAsFactors = FALSE)
for (col in panel_B_cols) {
  panel_B_table[[col]] <- character()
}

# Variables to extract
vars_to_extract <- c("INT_treatment", "TEMP_index", "TEMP_X_anytr_index")

# Fill Panel A
for (var in vars_to_extract) {
  coef_row <- data.frame(Variable = var, stringsAsFactors = FALSE)
  se_row <- data.frame(Variable = paste0("  ", var, "_se"), stringsAsFactors = FALSE)

  for (i in 1:length(panel_A_indices)) {
    model_idx <- panel_A_indices[i]
    col_name <- panel_A_cols[i]

    if (model_idx <= length(models_list) && !is.null(models_list[[model_idx]])) {
      results <- extract_model_results(models_list[[model_idx]])
      coef_row[[col_name]] <- if (!is.null(results[[var]])) results[[var]]$coef_formatted else "NA"
      se_row[[col_name]] <- if (!is.null(results[[var]])) results[[var]]$se_formatted else "(NA)"
    } else {
      coef_row[[col_name]] <- "NA"
      se_row[[col_name]] <- "(NA)"
    }
  }

  panel_A_table <- rbind(panel_A_table, coef_row, se_row)
}

# Fill Panel B
for (var in vars_to_extract) {
  coef_row <- data.frame(Variable = var, stringsAsFactors = FALSE)
  se_row <- data.frame(Variable = paste0("  ", var, "_se"), stringsAsFactors = FALSE)

  for (i in 1:length(panel_B_indices)) {
    model_idx <- panel_B_indices[i]
    col_name <- panel_B_cols[i]

    if (model_idx <= length(models_list) && !is.null(models_list[[model_idx]])) {
      results <- extract_model_results(models_list[[model_idx]])
      coef_row[[col_name]] <- if (!is.null(results[[var]])) results[[var]]$coef_formatted else "NA"
      se_row[[col_name]] <- if (!is.null(results[[var]])) results[[var]]$se_formatted else "(NA)"
    } else {
      coef_row[[col_name]] <- "NA"
      se_row[[col_name]] <- "(NA)"
    }
  }

  panel_B_table <- rbind(panel_B_table, coef_row, se_row)
}

# Add additional statistics to Panel A
obs_row_A <- data.frame(Variable = "Observations", stringsAsFactors = FALSE)
for (i in 1:length(panel_A_indices)) {
  model_idx <- panel_A_indices[i]
  col_name <- panel_A_cols[i]

  if (model_idx <= length(models_list) && !is.null(models_list[[model_idx]])) {
    obs_row_A[[col_name]] <- nobs(models_list[[model_idx]])
  } else {
    obs_row_A[[col_name]] <- "NA"
  }
}
panel_A_table <- rbind(panel_A_table, obs_row_A)

mean_row_A <- data.frame(Variable = "Mean in Control not WR in 2005", stringsAsFactors = FALSE)
for (i in 1:length(panel_A_indices)) {
  model_idx <- panel_A_indices[i]
  col_name <- panel_A_cols[i]
  mean_row_A[[col_name]] <- control_means[model_idx]
}
panel_A_table <- rbind(panel_A_table, mean_row_A)

test1_row_A <- data.frame(Variable = "Test Treat Effect", stringsAsFactors = FALSE)
test2_row_A <- data.frame(Variable = "Test Perf Effect in Treat", stringsAsFactors = FALSE)
for (i in 1:length(panel_A_indices)) {
  model_idx <- panel_A_indices[i]
  col_name <- panel_A_cols[i]
  test1_row_A[[col_name]] <- pvals_1[model_idx]
  test2_row_A[[col_name]] <- pvals_2[model_idx]
}
panel_A_table <- rbind(panel_A_table, test1_row_A, test2_row_A)

# Add additional statistics to Panel B
obs_row_B <- data.frame(Variable = "Observations", stringsAsFactors = FALSE)
for (i in 1:length(panel_B_indices)) {
  model_idx <- panel_B_indices[i]
  col_name <- panel_B_cols[i]

  if (model_idx <= length(models_list) && !is.null(models_list[[model_idx]])) {
    obs_row_B[[col_name]] <- nobs(models_list[[model_idx]])
  } else {
```

```r
      obs_row_B[[col_name]] <- "NA"
    }
  }
  panel_B_table <- rbind(panel_B_table, obs_row_B)

  mean_row_B <- data.frame(Variable = "Mean in Control not WR in 2005", stringsAsFactors = FALSE)
  for (i in 1:length(panel_B_indices)) {
    model_idx <- panel_B_indices[i]
    col_name <- panel_B_cols[i]
    mean_row_B[[col_name]] <- control_means[model_idx]
  }
  panel_B_table <- rbind(panel_B_table, mean_row_B)

  test1_row_B <- data.frame(Variable = "Test Treat Effect", stringsAsFactors = FALSE)
  test2_row_B <- data.frame(Variable = "Test Perf Effect in Treat", stringsAsFactors = FALSE)
  for (i in 1:length(panel_B_indices)) {
    model_idx <- panel_B_indices[i]
    col_name <- panel_B_cols[i]
    test1_row_B[[col_name]] <- pvals_1[model_idx]
    test2_row_B[[col_name]] <- pvals_2[model_idx]
  }
  panel_B_table <- rbind(panel_B_table, test1_row_B, test2_row_B)

  return(list(panel_A = panel_A_table, panel_B = panel_B_table))
}

# Create the reduced tables
reduced_tables <- create_reduced_performance_table(models_list, control_means, pvals_1, pvals_2)

# Display the tables
print("=== PANEL A: GP without Gender Quota in 2005 ===")
```

```
## [1] "=== PANEL A: GP without Gender Quota in 2005 ==="
```

```r
print(reduced_tables$panel_A)
```

```
##                          Variable  Model_1   Model_5
## 1                   INT_treatment -0.266** -6.761***
## 2                INT_treatment_se  (0.101)   (2.515)
## 3                      TEMP_index   -0.089   -3.031*
## 4                   TEMP_index_se  (0.069)   (1.713)
## 5              TEMP_X_anytr_index  0.235**   7.048**
## 6           TEMP_X_anytr_index_se  (0.114)    (2.82)
## 7                    Observations       92        90
## 8  Mean in Control not WR in 2005     0.46      10.1
## 9               Test Treat Effect     <NA>      <NA>
## 10      Test Perf Effect in Treat     <NA>      <NA>
##    Model_13 Model_17 Model_25 Model_29
## 1     0.066    0.597    0.104    1.441
## 2   (0.055)    (1.2)  (0.069)  (0.994)
## 3     0.014   -0.162   -0.016   -0.507
## 4   (0.037)  (0.817)  (0.047)  (0.677)
## 5    -0.055   -0.437   -0.018    1.053
## 6   (0.061)  (1.345)  (0.078)  (1.115)
## 7        92       90       92       90
## 8      0.04     0.73     0.08     0.86
## 9      <NA>     <NA>     <NA>     <NA>
## 10     <NA>     <NA>     <NA>     <NA>
```

```r
print("\n=== PANEL B: GP with Gender Quota in 2005 ===")
```

```
## [1] "\n=== PANEL B: GP with Gender Quota in 2005 ==="
```

```r
print(reduced_tables$panel_B)
```

```
##                           Variable Model_37 Model_41
## 1                    INT_treatment     0.15    2.479
## 2                 INT_treatment_se  (0.113)  (2.535)
## 3                       TEMP_index   -0.033   -0.182
## 4                    TEMP_index_se  (0.121)  (2.714)
## 5               TEMP_X_anytr_index   -0.133   -2.542
## 6            TEMP_X_anytr_index_se  (0.143)  (3.227)
## 7                     Observations       60       59
## 8  Mean in Control not WR in 2005     0.15      2.5
## 9                Test Treat Effect     <NA>     <NA>
## 10      Test Perf Effect in Treat     <NA>     <NA>
##    Model_49 Model_53 Model_61 Model_65
## 1  -0.268**   -5.815    0.123    2.493
## 2    (0.13)  (3.577)  (0.073)  (2.615)
## 3     0.066    4.566   -0.005     1.32
## 4    (0.14)  (3.829)  (0.078)    (2.8)
## 5     -0.09   -5.115   0.165*    1.288
## 6   (0.166)  (4.553)  (0.093)  (3.329)
## 7        60       59       60       59
## 8      0.33     6.15        0        0
## 9      <NA>     <NA>     <NA>     <NA>
## 10     <NA>     <NA>     <NA>     <NA>
```

Table 3 - Challengers 2010

Required libraries.

```r
library(dplyr)
library(fixest)
library(stargazer)
library(haven)
library(broom)
library(aod)
```

Defining the macros: controls, interest and dependent variables.

```r
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")

outregvar2 <- c("INT_treatment", "RES05_gender", "X_anytr_genderres05")

dep_vars <- c("ELEC10_nbcands", "CHAL_nbchal", "CHAL_prop_female",
              "CHAL_voteshare_female", "CHAL_prop_nongen", "CHAL_voteshare_nongen")
```

Data processing.

```r
data_path <- "~/work/Electoral data cleaned.dta"
data <- read_dta(data_path)
data_filtered <- data %>% filter(RES10_gender == 0, GP_tag == 1)
```

Function for the regressions.

```r
run_regression_analysis <- function(data_subset, subset_name) {
  cat(paste("\n=== ANALYZE FOR:", subset_name, "===\n"))
  cat("Number of observations:", nrow(data_subset), "\n")

  results_list <- list()

  for (i in seq_along(dep_vars)) {
    dep_var <- dep_vars[i]
    cat(paste(" Dependent variable:", dep_var, "\n"))

    if (!dep_var %in% names(data_subset)) {
      cat(paste("    ATTENTION: Variable", dep_var, "not found!\n"))
      next
    }

    control_subset <- data_subset %>% filter(INT_treatment == 0, RES05_gender == 0)
    control_mean <- if (nrow(control_subset) == 0) NA else control_subset %>%
      summarise(mean_val = mean(!!sym(dep_var), na.rm = TRUE)) %>%
      pull(mean_val) %>%
      round(2)

    cat(paste("    Control mean (non-previously gender-reserved):", control_mean, "\n"))

    available_controls <- gpcontrols[gpcontrols %in% names(data_subset)]
    reg_vars <- c(outregvar2, available_controls)
    reg_vars <- reg_vars[reg_vars %in% names(data_subset)]

    formula_str <- paste(dep_var, "~", paste(reg_vars, collapse = " + "), "+ factor(district)")

    tryCatch({
      model <- lm(as.formula(formula_str), data = data_subset)
      coef_names <- names(coef(model))

      res05_coef <- coef_names[grepl("RES05_gender", coef_names)]
      anytr_coef <- coef_names[grepl("X_anytr_genderres05", coef_names)]

      if (length(res05_coef) > 0 && length(anytr_coef) > 0) {
        L <- matrix(0, nrow = 1, ncol = length(coef(model)))
        names_L <- names(coef(model))
        L[1, which(names_L == res05_coef[1])] <- 1
        L[1, which(names_L == anytr_coef[1])] <- 1

        restriction <- L %*% coef(model)
        var_restriction <- L %*% vcov(model) %*% t(L)
        wald_stat <- as.numeric(restriction^2 / var_restriction)
        pval <- round(1 - pchisq(wald_stat, df = 1), 3)
      } else {
        pval <- NA
      }

      results_list[[i]] <- list(
        model = model,
        dep_var = dep_var,
        control_mean = control_mean,
        joint_test_pval = pval,
        subset = subset_name,
        formula = formula_str,
        n_obs = nrow(model$model)
      )

    }, error = function(e) {
      cat(paste("    ERROR in the regression:", e$message, "\n"))
      results_list[[i]] <- NULL
    })
  }

  results_list <- results_list[!sapply(results_list, is.null)]
  return(results_list)
}
```

We then split the analysis into three sub-samples.

```
results_full <- run_regression_analysis(data_filtered, "Full Sample")
```

```
##
## === ANALYZE FOR: Full Sample ===
## Number of observations: 382
##  Dependent variable: ELEC10_nbcands
##     Control mean (non-previously gender-reserved): 7.39
##  Dependent variable: CHAL_nbchal
##     Control mean (non-previously gender-reserved): 7.14
##  Dependent variable: CHAL_prop_female
##     Control mean (non-previously gender-reserved): 0.13
##  Dependent variable: CHAL_voteshare_female
##     Control mean (non-previously gender-reserved): 12.89
##  Dependent variable: CHAL_prop_nongen
##     Control mean (non-previously gender-reserved): 0.79
##  Dependent variable: CHAL_voteshare_nongen
##     Control mean (non-previously gender-reserved): 78.7
```

```
data_inc_can_run <- data_filtered %>% filter(INC05_can_run == 1)
results_inc_can <- run_regression_analysis(data_inc_can_run, "Incumbent Can Run")
```

```
##
## === ANALYZE FOR: Incumbent Can Run ===
## Number of observations: 245
##  Dependent variable: ELEC10_nbcands
##     Control mean (non-previously gender-reserved): 7.36
##  Dependent variable: CHAL_nbchal
##     Control mean (non-previously gender-reserved): 6.96
##  Dependent variable: CHAL_prop_female
##     Control mean (non-previously gender-reserved): 0.11
##  Dependent variable: CHAL_voteshare_female
##     Control mean (non-previously gender-reserved): 10.74
##  Dependent variable: CHAL_prop_nongen
##     Control mean (non-previously gender-reserved): 0.66
##  Dependent variable: CHAL_voteshare_nongen
##     Control mean (non-previously gender-reserved): 66.04
```

```r
data_inc_cannot_run <- data_filtered %>% filter(INC05_can_run == 0)
results_inc_cannot <- run_regression_analysis(data_inc_cannot_run, "Incumbent Cannot Run")
```

```
##
## === ANALYZE FOR: Incumbent Cannot Run ===
## Number of observations: 137
##  Dependent variable: ELEC10_nbcands
##     Control mean (non-previously gender-reserved): 7.45
##  Dependent variable: CHAL_nbchal
##     Control mean (non-previously gender-reserved): 7.45
##  Dependent variable: CHAL_prop_female
##     Control mean (non-previously gender-reserved): 0.16
##  Dependent variable: CHAL_voteshare_female
##     Control mean (non-previously gender-reserved): 16.51
##  Dependent variable: CHAL_prop_nongen
##     Control mean (non-previously gender-reserved): 1
##  Dependent variable: CHAL_voteshare_nongen
##     Control mean (non-previously gender-reserved): 100
```

Output table into three panels, on the same .txt file.

```r
# Regression for each sample (each panel)
results_full <- run_regression_analysis(data_filtered, "Full Sample")
```

```
##
## === ANALYZE FOR: Full Sample ===
## Number of observations: 382
##  Dependent variable: ELEC10_nbcands
##     Control mean (non-previously gender-reserved): 7.39
##  Dependent variable: CHAL_nbchal
##     Control mean (non-previously gender-reserved): 7.14
##  Dependent variable: CHAL_prop_female
##     Control mean (non-previously gender-reserved): 0.13
##  Dependent variable: CHAL_voteshare_female
##     Control mean (non-previously gender-reserved): 12.89
##  Dependent variable: CHAL_prop_nongen
##     Control mean (non-previously gender-reserved): 0.79
##  Dependent variable: CHAL_voteshare_nongen
##     Control mean (non-previously gender-reserved): 78.7
```

```r
results_inc_can <- run_regression_analysis(data_inc_can_run, "Incumbent Can Run")
```

```
##
## === ANALYZE FOR: Incumbent Can Run ===
## Number of observations: 245
##  Dependent variable: ELEC10_nbcands
##     Control mean (non-previously gender-reserved): 7.36
##  Dependent variable: CHAL_nbchal
##     Control mean (non-previously gender-reserved): 6.96
##  Dependent variable: CHAL_prop_female
##     Control mean (non-previously gender-reserved): 0.11
##  Dependent variable: CHAL_voteshare_female
##     Control mean (non-previously gender-reserved): 10.74
##  Dependent variable: CHAL_prop_nongen
##     Control mean (non-previously gender-reserved): 0.66
##  Dependent variable: CHAL_voteshare_nongen
##     Control mean (non-previously gender-reserved): 66.04
```

```r
results_inc_cannot <- run_regression_analysis(data_inc_cannot_run, "Incumbent Cannot Run")
```

```
##
## === ANALYZE FOR: Incumbent Cannot Run ===
## Number of observations: 137
##  Dependent variable: ELEC10_nbcands
##     Control mean (non-previously gender-reserved): 7.45
##  Dependent variable: CHAL_nbchal
##     Control mean (non-previously gender-reserved): 7.45
##  Dependent variable: CHAL_prop_female
##     Control mean (non-previously gender-reserved): 0.16
##  Dependent variable: CHAL_voteshare_female
##     Control mean (non-previously gender-reserved): 16.51
##  Dependent variable: CHAL_prop_nongen
##     Control mean (non-previously gender-reserved): 1
##  Dependent variable: CHAL_voteshare_nongen
##     Control mean (non-previously gender-reserved): 100
```

```r
# Extract the results
panel_A_models <- lapply(results_full, function(x) x$model)
panel_B_models <- lapply(results_inc_can, function(x) x$model)
panel_C_models <- lapply(results_inc_cannot, function(x) x$model)

# Extract control means and p values for additional rows
control_means <- sapply(results_full, function(x) x$control_mean)
pvals <- sapply(results_full, function(x) x$joint_test_pval)

# Table for each panel, then combined into one txt file
panel_A <- stargazer(
  panel_A_models,
  type = "text",
  column.labels = c("Model 1", "Model 2", "Model 3", "Model 4", "Model 5", "Model 6"),
  keep = c("INT_treatment", "RES05_gender", "X_anytr_genderres05"),
  add.lines = list(
    c("District FE", rep("Yes", length(panel_A_models))),
    c("GP Controls", rep("Yes", length(panel_A_models))),
    c("Mean in Control not WR in 2005", control_means),
    c("Test Treat Effect", pvals)
  ),
  digits = 2,
  title = "Panel A: All GPs",
  single.row = TRUE
)
```

```
##
## Panel A: All GPs
## ====================================================================================================
##                                                                                           Dependent
##                                 ---------------------------------------------------------------------
##                                                                                                formul
##                                       Model 1                Model 2                Model 3
##                                         (1)                    (2)                    (3)
## ----------------------------------------------------------------------------------------------------
## INT_treatment                       0.31 (0.56)            0.38 (0.56)            0.01 (0.03)
## RES05_gender                        1.03* (0.58)          1.22** (0.58)           0.01 (0.03)
## X_anytr_genderres05                 -0.79 (0.93)           -0.90 (0.94)          -0.0004 (0.04)
## ----------------------------------------------------------------------------------------------------
## District FE                             Yes                    Yes                    Yes
## GP Controls                             Yes                    Yes                    Yes
## Mean in Control not WR in 2005          7.39                   7.14                   0.13
## Test Treat Effect                       0.76                  0.694                  0.793
## Observations                            382                    382                    382
## R2                                      0.18                   0.18                   0.11
## Adjusted R2                             0.14                   0.14                   0.06
## Residual Std. Error                3.95 (df = 361)        3.96 (df = 361)        0.19 (df = 361)
## F Statistic                    4.06*** (df = 20; 361) 4.08*** (df = 20; 361) 2.16*** (df = 20; 361) 2
## ====================================================================================================
## Note:
```

```r
panel_B <- stargazer(
  panel_B_models,
  type = "text",
  column.labels = c("Model 1", "Model 2", "Model 3", "Model 4", "Model 5", "Model 6"),
  keep = c("INT_treatment", "RES05_gender", "X_anytr_genderres05"),
  add.lines = list(
    c("District FE", rep("Yes", length(panel_B_models))),
    c("GP Controls", rep("Yes", length(panel_B_models))),
    c("Mean in Control not WR in 2005", control_means),
    c("Test Treat Effect", pvals)
  ),
  digits = 2,
  title = "Panel B: Incumbent Can Run",
  single.row = TRUE
)
```

```
##
## Panel B: Incumbent Can Run
## ====================================================================================================
##                                                                                           Dependent
##                                 ---------------------------------------------------------------------
##                                                                                               formula
##                                       Model 1                Model 2                Model 3
##                                         (1)                    (2)                    (3)
## ----------------------------------------------------------------------------------------------------
## INT_treatment                       0.83 (0.77)            0.99 (0.77)           0.07** (0.03)
## RES05_gender                        1.14 (0.70)           1.46** (0.70)           -0.01 (0.03)
## X_anytr_genderres05                 -1.27 (1.15)           -1.49 (1.15)           -0.05 (0.05)
## ----------------------------------------------------------------------------------------------------
## District FE                             Yes                    Yes                    Yes
## GP Controls                             Yes                    Yes                    Yes
## Mean in Control not WR in 2005          7.39                   7.14                   0.13
## Test Treat Effect                       0.76                  0.694                  0.793
## Observations                            245                    245                    245
## R2                                      0.21                   0.22                   0.16
## Adjusted R2                             0.14                   0.15                   0.09
## Residual Std. Error               3.91 (df = 224)        3.91 (df = 224)        0.18 (df = 224)
## F Statistic                    3.01*** (df = 20; 224) 3.16*** (df = 20; 224) 2.14*** (df = 20; 224) 3
## ====================================================================================================
## Note:
```

```r
panel_C <- stargazer(
  panel_C_models,
  type = "text",
  column.labels = c("Model 1", "Model 2", "Model 3", "Model 4", "Model 5", "Model 6"),
  keep = c("INT_treatment", "RES05_gender", "X_anytr_genderres05"),
  add.lines = list(
    c("District FE", rep("Yes", length(panel_C_models))),
    c("GP Controls", rep("Yes", length(panel_C_models))),
    c("Mean in Control not WR in 2005", control_means),
    c("Test Treat Effect", pvals)
  ),
  digits = 2,
  title = "Panel C: Incumbent Cannot Run",
  single.row = TRUE
)
```

```
##
## Panel C: Incumbent Cannot Run
## ====================================================================================================
##                                                                                     Dependent variabl
##                                --------------------------------------------------------------------
##                                                                                         formula_str
##                                       Model 1                Model 2               Model 3
##                                         (1)                    (2)                   (3)
## ----------------------------------------------------------------------------------------------------
## INT_treatment                       0.17 (0.90)            0.17 (0.90)           -0.03 (0.05)
## RES05_gender                        1.06 (1.05)            1.02 (1.05)           0.03 (0.05)
## X_anytr_genderres05                 -0.76 (1.77)           -0.72 (1.77)          0.002 (0.09)
## ----------------------------------------------------------------------------------------------------
## District FE                             Yes                    Yes                   Yes
## GP Controls                             Yes                    Yes                   Yes
## Mean in Control not WR in 2005          7.39                   7.14                  0.13
## Test Treat Effect                       0.76                  0.694                 0.793
## Observations                            137                    137                   137
## R2                                      0.29                   0.29                  0.20
## Adjusted R2                             0.17                   0.17                  0.07
## Residual Std. Error               3.94 (df = 117)        3.94 (df = 117)       0.20 (df = 117)     23
## F Statistic                    2.48*** (df = 19; 117) 2.48*** (df = 19; 117) 1.58* (df = 19; 117) 1.3
## ====================================================================================================
## Note:
```

```r
# Combine three panels
combined_output <- c(panel_A, "\n\n", panel_B, "\n\n", panel_C)
writeLines(combined_output, "Table3_Challengers_2010_completed.txt")
```

Non-formatted table to fit in the PDF:

```r
create_outreg_table <- function(results_list_full, results_list_inc, results_list_no_inc) {
  extract_model_results <- function(model_result) {
```

```
  if (is.null(model_result) || is.null(model_result$model)) return(NULL)

  model <- model_result$model
  summary_model <- summary(model)
  coef_table <- summary_model$coefficients

  results <- list()
  for (var in outregvar2) {
    matching_vars <- rownames(coef_table)[grepl(var, rownames(coef_table))]
    if (length(matching_vars) > 0) {
      var_name <- matching_vars[1]
      coef_val <- coef_table[var_name, "Estimate"]
      se_val <- coef_table[var_name, "Std. Error"]
      pval <- coef_table[var_name, "Pr(>|t|)"]

      stars <- if (pval < 0.01) "***" else if (pval < 0.05) "**" else if (pval < 0.1) "*" else ""

      results[[var]] <- list(
        coef = round(coef_val, 3),
        se = round(se_val, 3),
        pval = pval,
        stars = stars,
        coef_formatted = paste0(round(coef_val, 3), stars),
        se_formatted = paste0("(", round(se_val, 3), ")")
      )
    } else {
      results[[var]] <- list(
        coef = NA,
        se = NA,
        pval = NA,
        stars = "",
        coef_formatted = "NA",
        se_formatted = "(NA)"
      )
    }
  }
  return(results)
}

final_table <- data.frame(
  Variable = character(),
  Full_Sample = character(),
  Inc_Can_Run = character(),
  Inc_Cannot_Run = character(),
  stringsAsFactors = FALSE
)

for (i in seq_along(dep_vars)) {
  dep_var <- dep_vars[i]
  full_results <- if (i <= length(results_list_full)) extract_model_results(results_list_full[[i]]) el
  inc_results <- if (i <= length(results_list_inc)) extract_model_results(results_list_inc[[i]]) else
  no_inc_results <- if (i <= length(results_list_no_inc)) extract_model_results(results_list_no_inc[[
```

```r
    for (var in outregvar2) {
      coef_row <- data.frame(
        Variable = var,
        Full_Sample = if (!is.null(full_results[[var]])) full_results[[var]]$coef_formatted else "NA",
        Inc_Can_Run = if (!is.null(inc_results[[var]])) inc_results[[var]]$coef_formatted else "NA",
        Inc_Cannot_Run = if (!is.null(no_inc_results[[var]])) no_inc_results[[var]]$coef_formatted else
        stringsAsFactors = FALSE
      )

      se_row <- data.frame(
        Variable = paste0("  ", var, "_se"),
        Full_Sample = if (!is.null(full_results[[var]])) full_results[[var]]$se_formatted else "(NA)",
        Inc_Can_Run = if (!is.null(inc_results[[var]])) inc_results[[var]]$se_formatted else "(NA)",
        Inc_Cannot_Run = if (!is.null(no_inc_results[[var]])) no_inc_results[[var]]$se_formatted else "
        stringsAsFactors = FALSE
      )

      final_table <- rbind(final_table, coef_row, se_row)
    }
  }

  return(final_table)
}

main_table <- create_outreg_table(results_full, results_inc_can, results_inc_cannot)
print(main_table)
```

```
##                    Variable Full_Sample Inc_Can_Run
## 1             INT_treatment       0.305       0.835
## 2          INT_treatment_se      (0.56)     (0.768)
## 3              RES05_gender      1.035*       1.143
## 4           RES05_gender_se     (0.575)     (0.704)
## 5       X_anytr_genderres05      -0.789       -1.27
## 6    X_anytr_genderres05_se     (0.933)     (1.155)
## 7             INT_treatment       0.385       0.986
## 8          INT_treatment_se     (0.562)     (0.768)
## 9              RES05_gender     1.221**     1.459**
## 10          RES05_gender_se     (0.577)     (0.704)
## 11      X_anytr_genderres05      -0.904       -1.49
## 12   X_anytr_genderres05_se     (0.935)     (1.154)
## 13            INT_treatment       0.006     0.068**
## 14         INT_treatment_se     (0.027)     (0.034)
## 15             RES05_gender        0.01      -0.007
## 16          RES05_gender_se     (0.027)     (0.032)
## 17      X_anytr_genderres05           0      -0.051
## 18   X_anytr_genderres05_se     (0.045)     (0.052)
## 19            INT_treatment      -2.805       2.447
## 20         INT_treatment_se     (2.966)     (3.668)
## 21             RES05_gender       4.089       3.466
## 22          RES05_gender_se     (3.012)     (3.336)
## 23      X_anytr_genderres05       0.534      -5.082
## 24   X_anytr_genderres05_se     (4.926)     (5.522)
## 25            INT_treatment     0.069**      0.095*
## 26         INT_treatment_se     (0.033)     (0.056)
## 27             RES05_gender       0.045      0.086*
## 28          RES05_gender_se     (0.034)     (0.051)
## 29      X_anytr_genderres05      -0.054       -0.07
## 30   X_anytr_genderres05_se     (0.055)     (0.084)
## 31            INT_treatment     7.488**     10.094*
## 32         INT_treatment_se     (3.539)     (5.914)
## 33             RES05_gender       4.088       8.289
## 34          RES05_gender_se     (3.594)     (5.379)
## 35      X_anytr_genderres05      -7.827      -10.16
## 36   X_anytr_genderres05_se     (5.877)     (8.904)
##    Inc_Cannot_Run
## 1           0.174
## 2         (0.901)
## 3            1.06
## 4         (1.047)
## 5          -0.759
## 6         (1.769)
## 7           0.174
## 8         (0.901)
## 9           1.023
## 10        (1.047)
## 11         -0.717
## 12        (1.769)
## 13         -0.033
## 14        (0.046)
## 15          0.032
## 16        (0.054)
## 17          0.002
## 18        (0.091)
## 19         -6.197
## 20         (5.45)
## 21          4.313
## 22         (6.26)
## 23          0.789
## 24       (10.645)
## 25              0
## 26        (0.008)
## 27         -0.016
## 28         (0.01)
## 29          0.018
## 30        (0.016)
## 31          0.004
## 32        (1.261)
## 33          -2.36
## 34        (1.449)
## 35          2.643
## 36        (2.464)
```

Table 4- Candidates 2015

Required libraries.

```
library(tidyverse)
library(fixest)
```

```r
library(stargazer)
library(haven)
```

Defining the macros: controls, regression variables.

```r
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")

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

Data processing: upload and filter.

```r
# 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))
}
```

Variables of the regression.

```r
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")
```

Lists to stock the results.

```r
models_list <- list()
control_means <- numeric(length(dep_vars))
pvals <- numeric(length(dep_vars))
```

Regressions.

```r
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 = " + "), "+ fa
  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$model))$coefficient
  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
```

Output table in .txt file.

```
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:
```

Non formatted table to fit into the PDF.

```
create_regression_table <- function(models_list, dep_vars, outregvar2) {
  extract_model_results <- function(model) {
    if (is.null(model)) return(NULL)

    summary_model <- summary(model)
    coef_table <- summary_model$coefficients

    results <- list()
    for (var in outregvar2) {
      if (var %in% rownames(coef_table)) {
        coef_val <- coef_table[var, "Estimate"]
        se_val <- coef_table[var, "Std. Error"]
        pval <- coef_table[var, "Pr(>|t|)"]

        stars <- if (pval < 0.01) "***" else if (pval < 0.05) "**" else if (pval < 0.1) "*" else ""

        results[[var]] <- list(
          coef = round(coef_val, 3),
          se = round(se_val, 3),
          pval = pval,
          stars = stars,
          coef_formatted = paste0(round(coef_val, 3), stars),
          se_formatted = paste0("(", round(se_val, 3), ")")
        )
      } else {
        results[[var]] <- list(
          coef = NA,
          se = NA,
          pval = NA,
          stars = "",
          coef_formatted = "NA",
          se_formatted = "(NA)"
        )
      }
    }
    return(results)
  }

  final_table <- data.frame(
    Variable = character(),
    stringsAsFactors = FALSE
  )

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

    for (var in outregvar2) {
      coef_row <- data.frame(
        Variable = var,
        stringsAsFactors = FALSE
      )

      se_row <- data.frame(
        Variable = paste0("  ", var, "_se"),
        stringsAsFactors = FALSE
      )

      for (j in seq_along(models_list)) {
        col_name <- paste0("Model_", j)
        coef_row[[col_name]] <- if (!is.null(model_results[[var]])) model_results[[var]]$coef_formatted
        se_row[[col_name]] <- if (!is.null(model_results[[var]])) model_results[[var]]$se_formatted els
      }

      final_table <- rbind(final_table, coef_row, se_row)
    }
  }

  return(final_table)
}

regression_table <- create_regression_table(models_list, dep_vars, outregvar2)
print(regression_table)
```

```
##                    Variable Model_1 Model_2 Model_3
## 1             INT_treatment   0.349   0.349   0.349
## 2          INT_treatment_se (1.214) (1.214) (1.214)
## 3              RES05_gender   0.818   0.818   0.818
## 4           RES05_gender_se (1.132) (1.132) (1.132)
## 5       X_anytr_genderres05   -2.575   -2.575   -2.575
## 6    X_anytr_genderres05_se   (1.89)   (1.89)   (1.89)
## 7             INT_treatment    0.069    0.069    0.069
## 8          INT_treatment_se  (0.049)  (0.049)  (0.049)
## 9              RES05_gender 0.122*** 0.122*** 0.122***
## 10          RES05_gender_se  (0.045)  (0.045)  (0.045)
## 11      X_anytr_genderres05  -0.136*  -0.136*  -0.136*
## 12   X_anytr_genderres05_se  (0.076)  (0.076)  (0.076)
## 13            INT_treatment  2.625**  2.625**  2.625**
## 14         INT_treatment_se  (1.286)  (1.286)  (1.286)
## 15             RES05_gender  2.424**  2.424**  2.424**
## 16          RES05_gender_se  (1.198)  (1.198)  (1.198)
## 17      X_anytr_genderres05  -3.715*  -3.715*  -3.715*
## 18   X_anytr_genderres05_se  (2.001)  (2.001)  (2.001)
## 19            INT_treatment    0.005    0.005    0.005
## 20         INT_treatment_se  (0.021)  (0.021)  (0.021)
## 21             RES05_gender   -0.014   -0.014   -0.014
## 22          RES05_gender_se   (0.02)   (0.02)   (0.02)
## 23      X_anytr_genderres05    0.012    0.012    0.012
## 24   X_anytr_genderres05_se  (0.033)  (0.033)  (0.033)
## 25            INT_treatment    0.678    0.678    0.678
## 26         INT_treatment_se  (3.008)  (3.008)  (3.008)
## 27             RES05_gender   -2.227   -2.227   -2.227
## 28          RES05_gender_se  (2.804)  (2.804)  (2.804)
## 29      X_anytr_genderres05    2.976    2.976    2.976
## 30   X_anytr_genderres05_se  (4.682)  (4.682)  (4.682)
## 31            INT_treatment  0.183**  0.183**  0.183**
## 32         INT_treatment_se  (0.083)  (0.083)  (0.083)
## 33             RES05_gender    0.025    0.025    0.025
## 34          RES05_gender_se  (0.077)  (0.077)  (0.077)
## 35      X_anytr_genderres05   -0.059   -0.059   -0.059
## 36   X_anytr_genderres05_se  (0.129)  (0.129)  (0.129)
## 37            INT_treatment  17.391*  17.391*  17.391*
## 38         INT_treatment_se  (8.859)  (8.859)  (8.859)
## 39             RES05_gender    7.566    7.566    7.566
## 40          RES05_gender_se  (8.259)  (8.259)  (8.259)
## 41      X_anytr_genderres05  -11.201  -11.201  -11.201
## 42   X_anytr_genderres05_se (13.792) (13.792) (13.792)
## 43            INT_treatment    0.087    0.087    0.087
## 44         INT_treatment_se  (0.078)  (0.078)  (0.078)
## 45             RES05_gender   -0.086   -0.086   -0.086
## 46          RES05_gender_se  (0.073)  (0.073)  (0.073)
## 47      X_anytr_genderres05    0.148    0.148    0.148
## 48   X_anytr_genderres05_se  (0.122)  (0.122)  (0.122)
## 49            INT_treatment    6.688    6.688    6.688
## 50         INT_treatment_se  (8.328)  (8.328)  (8.328)
## 51             RES05_gender   -7.176   -7.176   -7.176
## 52          RES05_gender_se  (7.764)  (7.764)  (7.764)
## 53      X_anytr_genderres05    9.024    9.024    9.024
## 54   X_anytr_genderres05_se (12.965) (12.965) (12.965)
##      Model_4  Model_5  Model_6  Model_7  Model_8  Model_9
## 1      0.349    0.349    0.349    0.349    0.349    0.349
## 2    (1.214)  (1.214)  (1.214)  (1.214)  (1.214)  (1.214)
## 3      0.818    0.818    0.818    0.818    0.818    0.818
## 4   (1.132)  (1.132)  (1.132)  (1.132)  (1.132)  (1.132)
## 5    -2.575   -2.575   -2.575   -2.575   -2.575   -2.575
## 6    (1.89)   (1.89)   (1.89)   (1.89)   (1.89)   (1.89)
## 7     0.069    0.069    0.069    0.069    0.069    0.069
## 8   (0.049)  (0.049)  (0.049)  (0.049)  (0.049)  (0.049)
## 9  0.122*** 0.122*** 0.122*** 0.122*** 0.122*** 0.122***
## 10  (0.045)  (0.045)  (0.045)  (0.045)  (0.045)  (0.045)
## 11  -0.136*  -0.136*  -0.136*  -0.136*  -0.136*  -0.136*
## 12  (0.076)  (0.076)  (0.076)  (0.076)  (0.076)  (0.076)
## 13  2.625**  2.625**  2.625**  2.625**  2.625**  2.625**
## 14  (1.286)  (1.286)  (1.286)  (1.286)  (1.286)  (1.286)
## 15  2.424**  2.424**  2.424**  2.424**  2.424**  2.424**
## 16  (1.198)  (1.198)  (1.198)  (1.198)  (1.198)  (1.198)
## 17  -3.715*  -3.715*  -3.715*  -3.715*  -3.715*  -3.715*
## 18  (2.001)  (2.001)  (2.001)  (2.001)  (2.001)  (2.001)
## 19    0.005    0.005    0.005    0.005    0.005    0.005
## 20  (0.021)  (0.021)  (0.021)  (0.021)  (0.021)  (0.021)
## 21   -0.014   -0.014   -0.014   -0.014   -0.014   -0.014
## 22   (0.02)   (0.02)   (0.02)   (0.02)   (0.02)   (0.02)
## 23    0.012    0.012    0.012    0.012    0.012    0.012
## 24  (0.033)  (0.033)  (0.033)  (0.033)  (0.033)  (0.033)
## 25    0.678    0.678    0.678    0.678    0.678    0.678
## 26  (3.008)  (3.008)  (3.008)  (3.008)  (3.008)  (3.008)
## 27   -2.227   -2.227   -2.227   -2.227   -2.227   -2.227
## 28  (2.804)  (2.804)  (2.804)  (2.804)  (2.804)  (2.804)
## 29    2.976    2.976    2.976    2.976    2.976    2.976
## 30  (4.682)  (4.682)  (4.682)  (4.682)  (4.682)  (4.682)
## 31  0.183**  0.183**  0.183**  0.183**  0.183**  0.183**
## 32  (0.083)  (0.083)  (0.083)  (0.083)  (0.083)  (0.083)
## 33    0.025    0.025    0.025    0.025    0.025    0.025
## 34  (0.077)  (0.077)  (0.077)  (0.077)  (0.077)  (0.077)
## 35   -0.059   -0.059   -0.059   -0.059   -0.059   -0.059
## 36  (0.129)  (0.129)  (0.129)  (0.129)  (0.129)  (0.129)
## 37  17.391*  17.391*  17.391*  17.391*  17.391*  17.391*
## 38  (8.859)  (8.859)  (8.859)  (8.859)  (8.859)  (8.859)
## 39    7.566    7.566    7.566    7.566    7.566    7.566
## 40  (8.259)  (8.259)  (8.259)  (8.259)  (8.259)  (8.259)
## 41  -11.201  -11.201  -11.201  -11.201  -11.201  -11.201
## 42 (13.792) (13.792) (13.792) (13.792) (13.792) (13.792)
## 43    0.087    0.087    0.087    0.087    0.087    0.087
## 44  (0.078)  (0.078)  (0.078)  (0.078)  (0.078)  (0.078)
## 45   -0.086   -0.086   -0.086   -0.086   -0.086   -0.086
## 46  (0.073)  (0.073)  (0.073)  (0.073)  (0.073)  (0.073)
## 47    0.148    0.148    0.148    0.148    0.148    0.148
## 48  (0.122)  (0.122)  (0.122)  (0.122)  (0.122)  (0.122)
## 49    6.688    6.688    6.688    6.688    6.688    6.688
## 50  (8.328)  (8.328)  (8.328)  (8.328)  (8.328)  (8.328)
## 51   -7.176   -7.176   -7.176   -7.176   -7.176   -7.176
## 52  (7.764)  (7.764)  (7.764)  (7.764)  (7.764)  (7.764)
## 53    9.024    9.024    9.024    9.024    9.024    9.024
## 54 (12.965) (12.965) (12.965) (12.965) (12.965) (12.965)
```

Table 5 - Voters perception