Section 4: Election Surveys

Milan Svolik

4/20/2023

Load the merged survey data

- Two proprietary surveys conducted by the survey agencies KONDA and SONAR

```
load("data/df_surveys.RData")
```

Table 2

- Election surveys: The joint distribution of March and June outcomes
- weight_s5: survey weights post-stratified to match 2019 administrative population totals for the joint distribution of age, gender, and education as well as the margins from the 2018 legislative and the March and June 2019 mayoral elections

```
df_analyze <- df_surveys %>%
  mutate(weight = weight_s5) %>%
  filter(!is.na(weight))

dim(df_analyze)
```

```
## [1] 5480   29
```

The joint distribution of March and June outcomes

```
df_P <- df_analyze %>%
  group_by(vote_march_3, vote_june_3) %>%
  summarise(n = n(), weight=mean(weight)) %>%
  group_by(vote_march_3, vote_june_3) %>%
  mutate(n_weighted = n*weight) %>%
  ungroup() %>%
  mutate(
    p=n_weighted/sum(n_weighted),
    percentage=round(100*p, 2))
```

```
## `summarise()` has grouped output by 'vote_march_3'. You can override using the `.groups` argument.
```

```
df_P
```

```
## # A tibble: 9 x 7
##   vote_march_3 vote_june_3     n weight n_weighted       p percentage
##   <fct>        <fct>       <int>  <dbl>      <dbl>   <dbl>      <dbl>
## 1 AKP          AKP          2397  1520.   3643877. 0.345        34.5
## 2 AKP          CHP           218  1345.    293130. 0.0277        2.77
## 3 AKP          ABS            23  9513.    218799. 0.0207        2.07
## 4 CHP          AKP            47  2056.     96613. 0.00914       0.91
```

```
## 5 CHP          CHP          2594  1557.   4039594. 0.382        38.2
## 6 CHP          ABS             3 11109.     33328. 0.00315       0.32
## 7 ABS          AKP            39  5015.    195578. 0.0185        1.85
## 8 ABS          CHP            88  4652.    409358. 0.0387        3.87
## 9 ABS          ABS            71 23100.   1640077. 0.155        15.5
```

The three mechanisms

```
switching <- df_P$p[2]-df_P$p[4]
backlash <- df_P$p[8]-df_P$p[6]
disenchantment <- df_P$p[3]-df_P$p[7]

round(100*switching, 2)
```

```
## [1] 1.86
```

```
round(100*backlash, 2)
```

```
## [1] 3.56
```

```
round(100*disenchantment, 2)
```

```
## [1] 0.22
```

Estimate the confidence intervals via the bootstrap

- use post-stratification weights as boostrap probabilities to propagate sampling uncertainty into the final estimates

```
boot_rep <- 10000
mechs_out <- c()

# USE POST-STRATIFICATION WEIGHTS AS BOOTSTRAP PROBABILITIES
df_toboot <- df_analyze %>%
  mutate(rid = seq(1:n())) %>%
  select(rid, vote_march_3, vote_june_3, weight)

# BOOTSTRAP
p_boot_save <- c()
for (i in 1:boot_rep) {

  # DRAW RIDS
  rid_boot <- sample(df_toboot$rid, replace=TRUE, prob=df_toboot$weight)

  # EXTRACT THE DATA BASED ON THE DRAW
  df_boot <- df_toboot[rid_boot,]

  # COMPUTE THE JOINT PMF
  p_boot <- table(df_boot$vote_march_3, df_boot$vote_june_3)/sum(table(df_boot$vote_march_3, df_boot$vo

  # TRANSFORM TO A VECTOR THAT IS A ROWWISE TRANSFORMATION OF THE MATRIX
  p_boot <- c(t(p_boot))

  # SAVE IT
  p_boot_save <- rbind(p_boot_save, p_boot)
}
```

Extract the bootstrap draws of the joint distribution of March and June outcomes and the mechanisms

```
switching_boot <- c()
backlash_boot <- c()
disenchantment_boot <- c()

for (j in 1:boot_rep) {
  P <- matrix(p_boot_save[j,], nrow=3, byrow=TRUE)

  switching <- P[1,2] - P[2,1]
  switching_boot <- rbind(switching_boot, switching)

  backlash <- P[3,2] - P[2,3]
  backlash_boot <- rbind(backlash_boot, backlash)

  disenchantment <- P[1,3] - P[3,1]
  disenchantment_boot <- rbind(disenchantment_boot, disenchantment)
}
```

Bootstrap estimates of the mean and the confidence intervals of the joint distribution of March and June outcomes

- These are the estimates reported in Table 2

```
p_boot_mean <- apply(p_boot_save, 2, mean, na.rm=TRUE)
p_boot_lower <- apply(p_boot_save, 2, quantile, probs=c(.025), na.rm=TRUE)
p_boot_upper <- apply(p_boot_save, 2, quantile, probs=c(.975), na.rm=TRUE)

P_boot_mean <- matrix(p_boot_mean, nrow=3, byrow=TRUE)
round(100*P_boot_mean, 2)
```

```
##       [,1]  [,2]  [,3]
## [1,] 34.47  2.78  2.07
## [2,]  0.91 38.21  0.32
## [3,]  1.84  3.87 15.53
```

```
P_boot_lower <- matrix(p_boot_lower, nrow=3, byrow=TRUE)
round(100*P_boot_lower, 2)
```

```
##       [,1]  [,2]  [,3]
## [1,] 33.25  2.35  1.70
## [2,]  0.66 36.93  0.18
## [3,]  1.50  3.36 14.56
```

```
P_boot_upper <- matrix(p_boot_upper, nrow=3, byrow=TRUE)
round(100*P_boot_upper, 2)
```

```
##       [,1]  [,2]  [,3]
## [1,] 35.71  3.21  2.46
## [2,]  1.19 39.47  0.47
## [3,]  2.21  4.40 16.50
```

Bootstrap estimates of the margins of the joint distribution of March and June outcomes

- These are the estimates reported in Table 2

```
round(100*rowSums(P_boot_mean), 2)
```

```
## [1] 39.32 39.44 21.24
```

```
round(100*colSums(P_boot_mean), 2)
```

```
## [1] 37.23 44.86 17.91
```

```
round(100*rowSums(P_boot_lower), 2)
```

```
## [1] 37.30 37.77 19.42
```

```
round(100*colSums(P_boot_lower), 2)
```

```
## [1] 35.40 42.65 16.44
```

```
round(100*rowSums(P_boot_upper), 2)
```

```
## [1] 41.39 41.13 23.10
```

```
round(100*colSums(P_boot_upper), 2)
```

```
## [1] 39.11 47.08 19.43
```

Bootstrap estimates of the mean and the confidence intervals of the three mechanisms

```
switching_boot_mean <- apply(switching_boot, 2, mean, na.rm=TRUE)
switching_boot_lower <- apply(switching_boot, 2, quantile, probs=c(.025), na.rm=TRUE)
switching_boot_upper <- apply(switching_boot, 2, quantile, probs=c(.975), na.rm=TRUE)

backlash_boot_mean <- apply(backlash_boot, 2, mean, na.rm=TRUE)
backlash_boot_lower <- apply(backlash_boot, 2, quantile, probs=c(.025), na.rm=TRUE)
backlash_boot_upper <- apply(backlash_boot, 2, quantile, probs=c(.975), na.rm=TRUE)

disenchantment_boot_mean <- apply(disenchantment_boot, 2, mean, na.rm=TRUE)
disenchantment_boot_lower <- apply(disenchantment_boot, 2, quantile, probs=c(.025), na.rm=TRUE)
disenchantment_boot_upper <- apply(disenchantment_boot, 2, quantile, probs=c(.975), na.rm=TRUE)

# CONSOLIDATE
switching_boot_out <- c(switching_boot_mean, switching_boot_lower, switching_boot_upper)
backlash_boot_out <- c(backlash_boot_mean, backlash_boot_lower, backlash_boot_upper)
disenchantment_boot_out <- c(disenchantment_boot_mean, disenchantment_boot_lower, disenchantment_boot_u

round(100*switching_boot_out, 2)
```

```
## [1] 1.86 1.37 2.37
```

```
round(100*backlash_boot_out, 2)
```

```
## [1] 3.55 3.03 4.09
```

```
round(100*disenchantment_boot_out, 2)
```

```
## [1]  0.23 -0.29  0.75
```

Figure 4

- Election surveys: The joint distribution of March and June outcomes

Contruct the plot data

- weight_s5: survey weights post-stratified to match 2019 administrative population totals for the joint distribution of age, gender, and education as well as the margins from the 2018 legislative and the March and June 2019 mayoral elections

```
df_analyze <- df_surveys %>%
  mutate(weight = weight_s5) %>%
  filter(!is.na(weight))
```

A FUNCTION THAT CONDUCTS THE BOOTSTRAP

```
mech_boot_fn <- function(data) {

  boot_rep <- 10000

  # BOOTSTRAP
  p_boot_save <- c()
  for (i in 1:boot_rep) {

    # DRAW RIDS
    rid_boot <- sample(data$rid, replace=TRUE, prob=data$weight)

    # EXTRACT THE DATA BASED ON THE DRAW
    df_boot <- data[rid_boot,]

    # COMPUTE THE JOINT PMF
    p_boot <- table(df_boot$vote_march_3, df_boot$vote_june_3)/sum(table(df_boot$vote_march_3, df_boot$

    # TRANSFORM TO A VECTOR THAT IS A ROWWISE TRANSFORMATION OF THE MATRIX
    p_boot <- c(t(p_boot))

    # SAVE IT
    p_boot_save <- rbind(p_boot_save, p_boot)
  }

  # EXTRACT THE BOOTSTRAP DRAWS OF THE JOINT DISTRIBUTION OF MARCH AND JUNE OUTCOMES AND THE MECHANISMS
  switching_boot <- c()
  backlash_boot <- c()
  disenchantment_boot <- c()

  for (j in 1:boot_rep) {
    P <- matrix(p_boot_save[j,], nrow=3, byrow=TRUE)

    switching <- P[1,2] - P[2,1]
    switching_boot <- rbind(switching_boot, switching)

    backlash <- P[3,2] - P[2,3]
    backlash_boot <- rbind(backlash_boot, backlash)
```

```
    disenchantment <- P[1,3] - P[3,1]
    disenchantment_boot <- rbind(disenchantment_boot, disenchantment)
  }

  ## BOOTSTRAP ESTIMATES OF THE MEAN AND THE CONFIDENCE INTERVALS OF THE THREE MECHANISMS
  switching_boot_mean <- apply(switching_boot, 2, mean, na.rm=TRUE)
  switching_boot_lower <- apply(switching_boot, 2, quantile, probs=c(.025), na.rm=TRUE)
  switching_boot_upper <- apply(switching_boot, 2, quantile, probs=c(.975), na.rm=TRUE)

  backlash_boot_mean <- apply(backlash_boot, 2, mean, na.rm=TRUE)
  backlash_boot_lower <- apply(backlash_boot, 2, quantile, probs=c(.025), na.rm=TRUE)
  backlash_boot_upper <- apply(backlash_boot, 2, quantile, probs=c(.975), na.rm=TRUE)

  disenchantment_boot_mean <- apply(disenchantment_boot, 2, mean, na.rm=TRUE)
  disenchantment_boot_lower <- apply(disenchantment_boot, 2, quantile, probs=c(.025), na.rm=TRUE)
  disenchantment_boot_upper <- apply(disenchantment_boot, 2, quantile, probs=c(.975), na.rm=TRUE)

  # CONSOLIDATE
  switching_boot_out <- c(switching_boot_mean, switching_boot_lower, switching_boot_upper)
  backlash_boot_out <- c(backlash_boot_mean, backlash_boot_lower, backlash_boot_upper)
  disenchantment_boot_out <- c(disenchantment_boot_mean, disenchantment_boot_lower, disenchantment_boot

  return(list(switching_boot_out, backlash_boot_out, disenchantment_boot_out))
}
```

FILTER TO 2018 CHP VOTERS

```
# USE POST-STRATIFICATION WEIGHTS AS BOOTSTRAP PROBABILITIES
df_toboot <- df_analyze %>%
  filter(vote_2018=="CHP") %>%
  mutate(rid = seq(1:n())) %>%
  select(rid, vote_march_3, vote_june_3, weight)

# BOOTSTRAP
boot_out_list <- mech_boot_fn(df_toboot)

# SAVE FOR PLOTING
df_plot_CHP <- data.frame(rbind(
  c(-1, 1, boot_out_list[[1]]),
  c(-1, 2, boot_out_list[[2]]),
  c(-1, 3, boot_out_list[[3]])))
```

FILTER TO 2018 AKP VOTERS

```
# USE POST-STRATIFICATION WEIGHTS AS BOOTSTRAP PROBABILITIES
df_toboot <- df_analyze %>%
  filter(vote_2018=="AKP") %>%
  mutate(rid = seq(1:n())) %>%
  select(rid, vote_march_3, vote_june_3, weight)

# BOOTSTRAP
boot_out_list <- mech_boot_fn(df_toboot)
```

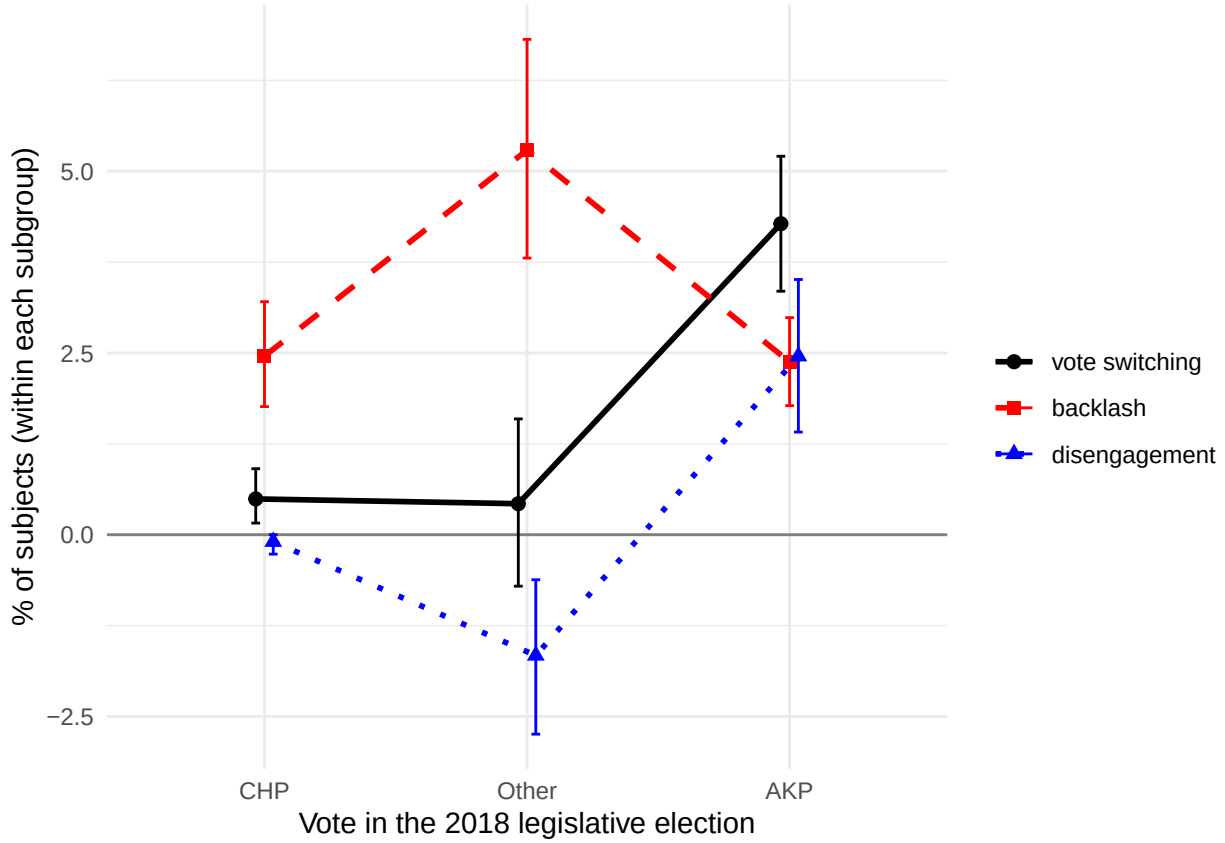
```
# SAVE FOR PLOTING
df_plot_AKP <- data.frame(rbind(
  c(1, 1, boot_out_list[[1]]),
  c(1, 2, boot_out_list[[2]]),
  c(1, 3, boot_out_list[[3]])))
```

FILTER TO 2018 NEITHER AKP OR CHP

```
# USE POST-STRATIFICATION WEIGHTS AS BOOTSTRAP PROBABILITIES
df_toboot <- df_analyze %>%
  filter(vote_2018!="AKP" & vote_2018!="CHP") %>%
  mutate(rid = seq(1:n())) %>%
  select(rid, vote_march_3, vote_june_3, weight)

# BOOTSTRAP
boot_out_list <- mech_boot_fn(df_toboot)

# SAVE FOR PLOTING
df_plot_OTHER <- data.frame(rbind(
  c(0, 1, boot_out_list[[1]]),
  c(0, 2, boot_out_list[[2]]),
  c(0, 3, boot_out_list[[3]])))
```

PLOT

```
df_plot <- rbind(df_plot_CHP, df_plot_OTHER, df_plot_AKP)
colnames(df_plot) <- c("vote2018", "mech", "estimate", "ci_lower", "ci_upper")

df_plot <- df_plot %>%
    mutate(mech=as.factor(mech)) %>%
    mutate(vote2018=as.factor(vote2018))

plot_group_labels <- c("vote switching", "backlash", "disengagement")
plot_x_label <- c("Vote in the 2018 legislative election")
plot_x_labels <- c("CHP", "Other", "AKP")
plot_x_dodge <- .1
df_plot %>%
  ggplot(aes(x=vote2018, y=100*estimate), col=mech) +
  geom_hline(yintercept=0, col="grey50") +
  geom_line(aes(col=mech, group=mech, linetype = mech),  size=1,  position = position_dodge(plot_x_dodg
  geom_errorbar(aes(ymin=100*ci_lower, ymax=100*ci_upper, col=mech), width=0.1,  position = position_do
  geom_point(aes(col=mech, shape=mech),  size=2, position = position_dodge(plot_x_dodge)) +
  scale_shape_manual(name  ="", values=c(19,15,17), labels=plot_group_labels) +
  scale_linetype_manual(name  ="", values=c("solid", "dashed", "dotted"), labels=plot_group_labels)+
  scale_color_manual(name  ="", values=c("black", "red", "blue"), labels=plot_group_labels) +
  theme_minimal() +
  scale_x_discrete(name = plot_x_label,
        labels=plot_x_labels) +
  scale_y_continuous(name = "% of subjects (within each subgroup)")
```




```
# SAVE THE PLOT
ggsave("figures/figure_4.png", dpi = 1200, width = 10, height = 6, units = "in")
ggsave("figures/figure_4.eps", dpi = 1200, width = 10, height = 6, units = "in")
ggsave("figures/figure_4.pdf", dpi = 1200, width = 10, height = 6, units = "in")
```

Table 3

- ELECTION SURVEYS: HETEROGENEITY IN PUNISHMENT BY RESPONDENT COVARIATES
- Just like earlier we are working with vote_3, which implies that:
  - third-party voters are treated as abstainers
  - I am dropping the undecided
- weight_s5: survey weights post-stratified to match 2019 administrative population totals for the joint distribution of age, gender, and education as well as the margins from the 2018 legislative and the March and June 2019 mayoral elections

```
rm(list = ls(all = TRUE))
load("data/df_surveys.RData")
```

PREPARE THE DATA FOR ESTIMATION

```
# SELECT THE VARIABLES TO BE USED
df_temp <- df_surveys %>%
  mutate(weight = weight_s5) %>%
  filter(!is.na(weight)) %>%
  select(vote_march_3, vote_june_3, vote_2018, dg_sex, dg_age, dg_edu, turkish, istanbul_born, weight, i
```

```
colnames(df_temp)
```

```
##  [1] "vote_march_3"  "vote_june_3"   "vote_2018"     "dg_sex"
##  [5] "dg_age"        "dg_edu"        "turkish"       "istanbul_born"
##  [9] "weight"        "ilce_name"
```

```
dim(df_temp)
```

```
## [1] 5480   10
```

```
# RECODE AGE TO HAVE THE SAME CATEGORIES AS IN THE SURVEY EXPERIMENTS
table(df_temp$dg_age)
```

```
##
## 18-24 25-34 35-44 45-54 55-64 65-up
##   751  1182  1347  1090   728   382
```

```
df_temp$dg_age <-  fct_recode(df_temp$dg_age,
                          `18-34` = "18-24",
                          `18-34` = "25-34",
                          `35-54` = "35-44",
                          `35-54` = "45-54",
                          `55+` = "55-64",
                          `55+` = "65-up")
table(df_temp$dg_age)
```

```
##
## 18-34 35-54   55+
##  1933  2437  1110
```

VOTE SWITCHING

- THE SAMPLE: all who voted for the AKP in March
  - MARCH: 0 if the subject voted for the AKP in March, NA otherwise
  - JUNE: 1 if the subject voted for the AKP in March but voted CHP in June; 0 otherwise
- THE DV y: the MARCH to JUNE shift
  - 0 if none
  - 1 if yes

```
# DV
df_vote_switching <- df_temp
table(df_vote_switching$vote_march_3, df_vote_switching$vote_june_3)
```

```
##
##        AKP  CHP  ABS
##   AKP 2397  218   23
##   CHP   47 2594    3
##   ABS   39   88   71
```

```
vote_march <- case_when(df_vote_switching$vote_march_3=="AKP" ~ 0)
vote_june <- case_when(df_vote_switching$vote_march_3=="AKP" & df_vote_switching$vote_june_3=="AKP" ~ 0
                           df_vote_switching$vote_march_3=="AKP" & df_vote_switching$vote_june_3=="CHP"

df_vote_switching <- df_vote_switching %>%
  mutate(vote_march, vote_june) %>%
  mutate(Y = vote_june - vote_march)
```

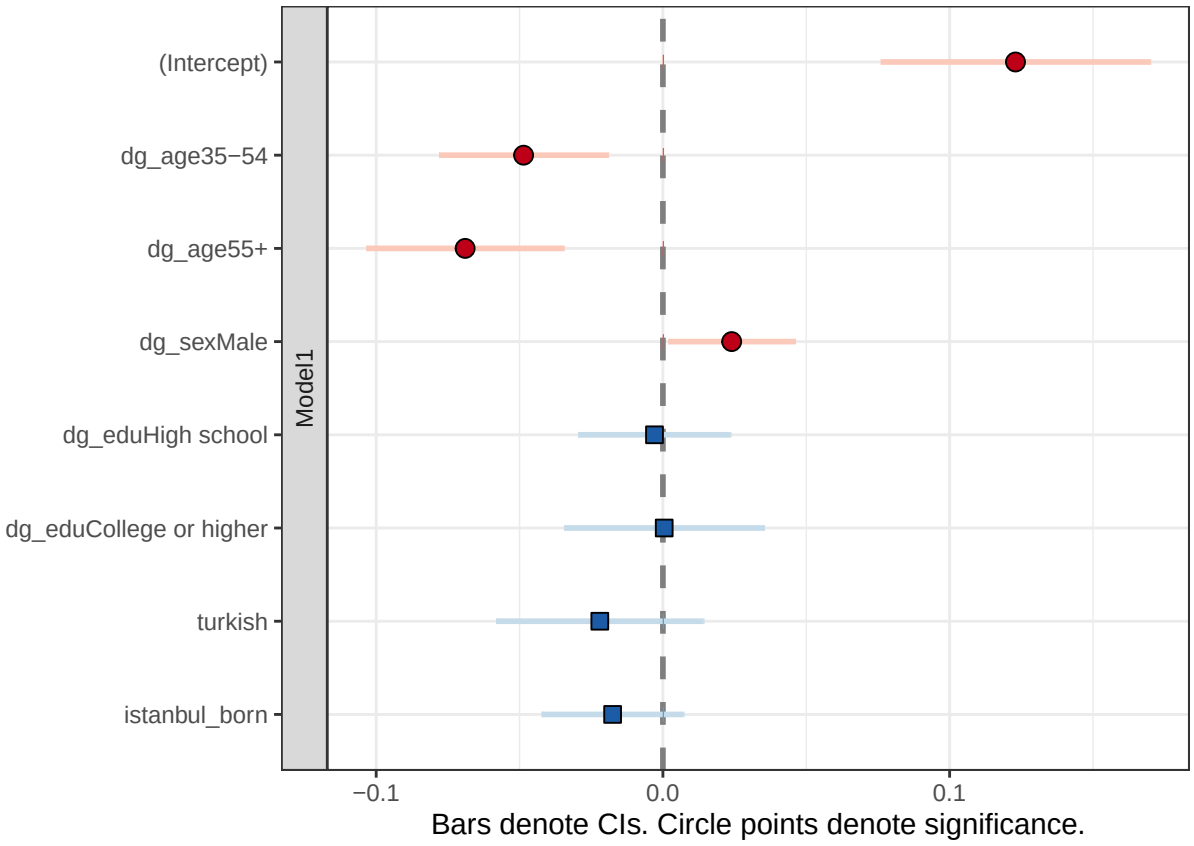
```
df_vote_switching %>%
  group_by(vote_march, vote_june) %>%
  summarise(N=n(), mean(Y))
```

```
## `summarise()` has grouped output by 'vote_march'. You can override using the `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   vote_march [2]
##   vote_march vote_june     N `mean(Y)`
##        <dbl>     <dbl> <int>     <dbl>
## 1          0         0  2397         0
## 2          0         1   218         1
## 3          0        NA    23        NA
## 4         NA        NA  2842        NA
```

```
df_ols <- df_vote_switching %>%
  select(Y, dg_age, dg_sex, dg_edu, turkish, istanbul_born, weight, ilce_name)

ols_vote_switching <- lm_robust(Y ~ .-weight-ilce_name, data=df_ols, weights = weight)
plotreg(list(ols_vote_switching))
```

```
## Model: bars denote 0.95 confidence intervals.
```



BACKLASH

- THE SAMPLE: ABS (including vote blank or vote third party) in March
  - MARCH: 0 if the subject ABS (abstained) in March, NA otherwise

    - JUNE: 1 if the subject ABS in March but voted CHP in June; 0 otherwise
- THE DV y: the MARCH to JUNE shift
    - 0 if none
    - 1 if yes

```
# DV
df_backlash <- df_temp
table(df_backlash$vote_march_3, df_backlash$vote_june_3)
```

```
##
##         AKP  CHP  ABS
##   AKP  2397  218   23
##   CHP    47 2594    3
##   ABS    39   88   71
```

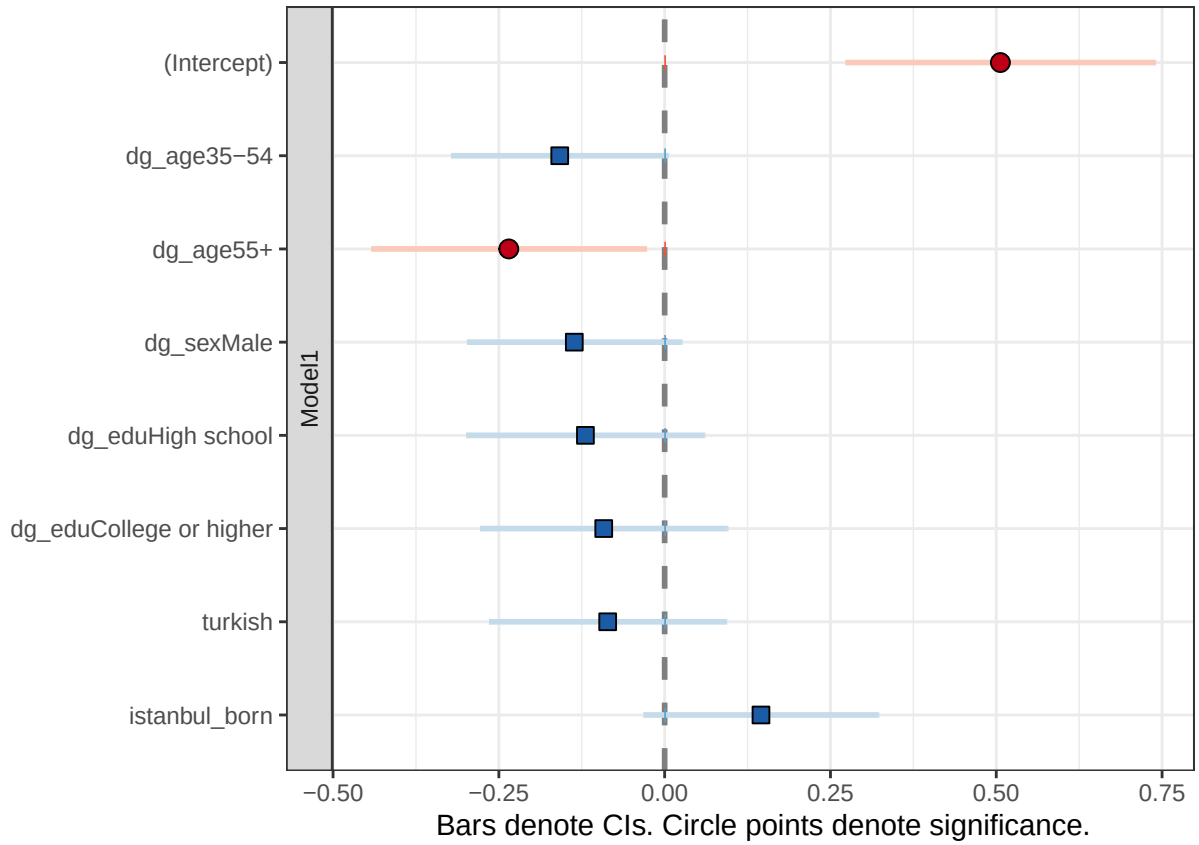
```
vote_march <- case_when(df_backlash$vote_march_3=="ABS" ~ 0)
vote_june <- case_when(df_backlash$vote_march_3=="ABS" & df_backlash$vote_june_3=="ABS" ~ 0,
                           df_backlash$vote_march_3=="ABS" & df_backlash$vote_june_3=="CHP" ~ 1)

df_backlash <- df_backlash %>%
  mutate(vote_march, vote_june) %>%
  mutate(Y = vote_june - vote_march)

df_backlash %>%
  group_by(vote_march, vote_june) %>%
  summarise(N=n(), mean(Y))
```

```
## `summarise()` has grouped output by 'vote_march'. You can override using the `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   vote_march [2]
##   vote_march vote_june     N `mean(Y)`
##        <dbl>     <dbl> <int>     <dbl>
## 1          0         0    71         0
## 2          0         1    88         1
## 3          0        NA    39        NA
## 4         NA        NA  5282        NA
```

```
df_ols <- df_backlash %>%
  select(Y, dg_age, dg_sex, dg_edu, turkish, istanbul_born, weight, ilce_name)

ols_backlash <- lm_robust(Y ~ .-weight-ilce_name, data=df_ols, weights = weight)
plotreg(list(ols_backlash))
```

```
## Model: bars denote 0.95 confidence intervals.
```



DISENCHANTMENT * THE SAMPLE: AKP in March + MARCH: 0 if the subject voted for the AKP, NA otherwise + JUNE: 1 if the subject voted for the AKP in March but ABS (abstained) in June; 0 otherwise * THE DV y: the MARCH to JUNE shift + 0 if none + 1 if yes

```
# DV
df_disenchantment <- df_temp
table(df_disenchantment$vote_march_3, df_disenchantment$vote_june_3)
```

```
##
##        AKP  CHP  ABS
##   AKP 2397  218   23
##   CHP   47 2594    3
##   ABS   39   88   71
```

```
vote_march <- case_when(df_disenchantment$vote_march_3=="AKP" ~ 0)
vote_june <- case_when(df_disenchantment$vote_march_3=="AKP" & df_disenchantment$vote_june_3=="AKP" ~ 0
                       df_disenchantment$vote_march_3=="AKP" & df_disenchantment$vote_june_3=="ABS"

df_disenchantment <- df_disenchantment %>%
  mutate(vote_march, vote_june) %>%
  mutate(Y = vote_june - vote_march)

df_disenchantment %>%
  group_by(vote_march, vote_june) %>%
  summarise(N=n(), mean(Y))
```

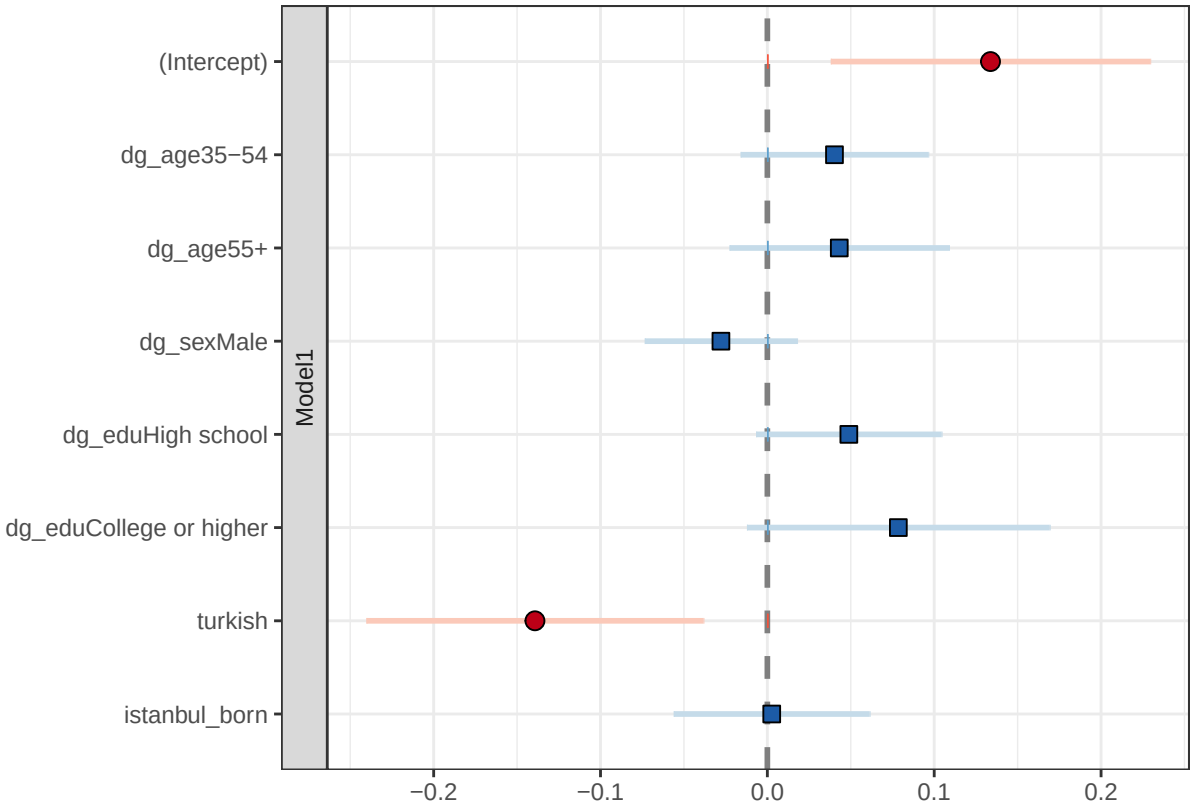
```
## `summarise()` has grouped output by 'vote_march'. You can override using the `.groups` argument.
```

```
## # A tibble: 4 x 4
```

```
## # Groups:   vote_march [2]
##   vote_march vote_june     N `mean(Y)`
##        <dbl>     <dbl> <int>     <dbl>
## 1          0         0  2397         0
## 2          0         1    23         1
## 3          0        NA   218        NA
## 4         NA        NA  2842        NA
```

```
df_ols <- df_disenchantment %>%
  select(Y, dg_age, dg_sex, dg_edu, turkish, istanbul_born, weight, ilce_name)

ols_disenchantment <- lm_robust(Y ~ .-weight-ilce_name, data=df_ols, weights = weight)
plotreg(list(ols_disenchantment))
```

```
## Model: bars denote 0.95 confidence intervals.
```



Bars denote CIs. Circle points denote significance.

Report Table 3

```
texreg(list(ols_vote_switching, ols_backlash, ols_disenchantment), include.ci = FALSE, digits = 3, custo
```

```
##
## \begin{table}
## \begin{center}
## \begin{tabular}{l c c c}
## \hline
##  & Vote Switching & Backlash & Disenchantment \\
## \hline
```

```
## Intercept                    & $0.123^{***}$  & $0.506^{***}$ & $0.134^{***}$  \\
##                              & $(0.024)$       & $(0.118)$     & $(0.049)$      \\
## Age: 35-54                   & $-0.049^{***}$ & $-0.158^{*}$  & $0.040$        \\
##                              & $(0.015)$       & $(0.083)$     & $(0.029)$      \\
## Age: 55+                     & $-0.069^{***}$ & $-0.235^{**}$ & $0.043$        \\
##                              & $(0.018)$       & $(0.105)$     & $(0.034)$      \\
## Sex: male                    & $0.024^{**}$    & $-0.136^{*}$  & $-0.028$       \\
##                              & $(0.011)$       & $(0.082)$     & $(0.023)$      \\
## Education: High school       & $-0.003$        & $-0.119$      & $0.049^{*}$    \\
##                              & $(0.014)$       & $(0.091)$     & $(0.028)$      \\
## Education: College or higher & $0.000$         & $-0.092$      & $0.079^{*}$    \\
##                              & $(0.018)$       & $(0.094)$     & $(0.046)$      \\
## Turkish                      & $-0.022$        & $-0.086$      & $-0.139^{***}$ \\
##                              & $(0.018)$       & $(0.090)$     & $(0.051)$      \\
## Istanbul born                & $-0.018$        & $0.145$       & $0.003$        \\
##                              & $(0.013)$       & $(0.090)$     & $(0.030)$      \\
## \hline
## R$^2$                        & $0.014$         & $0.100$       & $0.065$        \\
## Adj. R$^2$                   & $0.011$         & $0.057$       & $0.062$        \\
## Num. obs.                    & $2603$          & $155$         & $2409$         \\
## RMSE                         & $10.139$        & $44.160$      & $8.967$        \\
## \hline
## \multicolumn{4}{l}{\scriptsize{$^{***}p<0.01$; $^{**}p<0.05$; $^{*}p<0.1$}}
## \end{tabular}
## \caption{Statistical models}
## \label{table:coefficients}
## \end{center}
## \end{table}
```
‘‘‘