Section 3: The Istanbul Mayoral Election

Milan Svolik

4/20/2023

Load the data

```
load("data/df_2019istanbul_results.RData")
```

Summarize the number of districts, neighborhoods, and ballot boxes

```
df_results %>%
  distinct(district_id) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    39
```

```
df_results %>%
  distinct(neighborhood_id) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   956
```

```
df_results %>%
  distinct(box_id) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 31101
```

Aggregate shifts

- Third-party voters are treated as abstaining

```
df_results %>%
  mutate(
    ABS_march = ABS_march + THIRD_march,
    ABS_june = ABS_june + THIRD_june
    ) %>%
  summarize(AKP_march=sum(AKP_march), CHP_march=sum(CHP_march), ABS_march=sum(ABS_march), AKP_june=sum(A
  mutate(
```

```
    REG_march = AKP_march + CHP_march + ABS_march,
    AKP_share_march = AKP_march/REG_march,
    CHP_share_march = CHP_march/REG_march,
    ABS_share_march = ABS_march/REG_march,
    REG_june = AKP_june + CHP_june + ABS_june,
    AKP_share_june = AKP_june/REG_june,
    CHP_share_june = CHP_june/REG_june,
    ABS_share_june = ABS_june/REG_june,
    AKP_share_shift = AKP_share_june - AKP_share_march,
    CHP_share_shift = CHP_share_june - CHP_share_march,
    ABS_share_shift = ABS_share_june - ABS_share_march
  ) %>%
  select(AKP_share_shift, CHP_share_shift, ABS_share_shift) %>%
  round(4)
```

```
## # A tibble: 1 x 3
##   AKP_share_shift CHP_share_shift ABS_share_shift
##             <dbl>           <dbl>           <dbl>
## 1         -0.0208          0.0542         -0.0334
```

Figure 2

- ELECTION RESULTS: SHIFTS IN THE AKP TWO-PARTY VOTE SHARE, THE AKP VOTE, CHP VOTE, AND ABSTENTION BETWEEN MARCH AND JUNE 2019

Prepare the plot data

```
# CHECK ON THE SHARE OF THIRD-PARTY VOTERS
df_results %>%
  mutate(across(AKP_march:THIRD_june, sum)) %>%
  slice(1) %>%
  mutate(
    THIRD_share_march = THIRD_march/(AKP_march + CHP_march + ABS_march + THIRD_march),
    THIRD_share_june = THIRD_june/(AKP_june + CHP_june + ABS_june + THIRD_june)) %>%
  select(starts_with("THIRD"))
```

```
## # A tibble: 1 x 4
##   THIRD_march THIRD_june THIRD_share_march THIRD_share_june
##         <dbl>      <dbl>             <dbl>            <dbl>
## 1      223657      68318            0.0212          0.00647
```

```
# TREAT THIRD-PARTY VOTERS AS ABSTAINERS
df_results <- df_results %>%
  mutate(
    ABS_march = ABS_march + THIRD_march,
    ABS_june = ABS_june + THIRD_june
    ) %>%
  select(-THIRD_march, -THIRD_june)

# AGGREGATE THE RESULTS AT THE NEIGHBORHOOD LEVEL
df_results_neighbourhood <- df_results %>%
  group_by(district_id, district_name, neighborhood_id, neighborhood_name) %>%
  summarize(AKP_march=sum(AKP_march), CHP_march=sum(CHP_march), ABS_march=sum(ABS_march), AKP_june=sum(A
```

```
## `summarise()` has grouped output by 'district_id', 'district_name',
## 'neighborhood_id'. You can override using the `.groups` argument.
```

```
# AKP VOTE, CHP VOTE, ABSTENTIONS AS A SHARE OF REGISTERED VOTERS
df_results_neighbourhood <- df_results_neighbourhood %>%
  mutate(
    REG_march = AKP_march + CHP_march + ABS_march,
    AKP_share_march = AKP_march/REG_march,
    CHP_share_march = CHP_march/REG_march,
    ABS_share_march = ABS_march/REG_march,
    REG_june = AKP_june + CHP_june + ABS_june,
    AKP_share_june = AKP_june/REG_june,
    CHP_share_june = CHP_june/REG_june,
    ABS_share_june = ABS_june/REG_june,
  )

# SHIFTS IN AKP VOTE SHARE, CHP VOTE SHARE, AND ABSTENTION RATE
df_results_neighbourhood <- df_results_neighbourhood %>%
  mutate(
    AKP_share_diff = AKP_share_june - AKP_share_march,
    CHP_share_diff = CHP_share_june - CHP_share_march,
    ABS_share_diff = ABS_share_june - ABS_share_march
  )

# AKP TWO PARTY VOTE SHARE AND ITS SHIFT
df_results_neighbourhood <- df_results_neighbourhood %>%
  mutate(
    AKP_two_march = AKP_march/(AKP_march + CHP_march),
    AKP_two_june = AKP_june/(AKP_june + CHP_june),
    AKP_two_diff = AKP_two_june - AKP_two_march
  )

df_results_neighbourhood <- ungroup(df_results_neighbourhood)
```

Summarize turnout rates in March and June

```
df_results_neighbourhood %>%
  select(AKP_two_march, ABS_share_march, ABS_share_june) %>%
  summarize(turnout_march = mean(1-ABS_share_march), turnout_june = mean(1-ABS_share_june)) %>%
  round(4)
```

```
## # A tibble: 1 x 2
##   turnout_march turnout_june
##           <dbl>        <dbl>
## 1         0.796        0.822
```

```
df_results_neighbourhood %>%
  select(AKP_two_march, ABS_share_march, ABS_share_june) %>%
  group_by(pct=cut(AKP_two_march, c(0, seq(.2, .7, .1), 1))) %>%
  summarize(turnout_march = mean(1-ABS_share_march), turnout_june = mean(1-ABS_share_june))
```

```
## # A tibble: 7 x 3
##   pct       turnout_march turnout_june
##   <fct>             <dbl>        <dbl>
## 1 (0,0.2]           0.784        0.843
```

```
## 2 (0.2,0.3]         0.789        0.837
## 3 (0.3,0.4]         0.786        0.820
## 4 (0.4,0.5]         0.790        0.815
## 5 (0.5,0.6]         0.795        0.818
## 6 (0.6,0.7]         0.799        0.820
## 7 (0.7,1]           0.833        0.838
```

What do the groups along the horizontal axis look like?

```
df_results_neighbourhood %>%
  select(AKP_two_march) %>%
  summarize(min(AKP_two_march), max(AKP_two_march)) %>%
  round(digits = 3)
```

```
## # A tibble: 1 x 2
##   `min(AKP_two_march)` `max(AKP_two_march)`
##                  <dbl>                <dbl>
## 1                0.118                0.964
```

```
df_results_neighbourhood %>%
  select(AKP_two_march) %>%
  group_by(pct=cut(AKP_two_march, c(0, seq(.2, .7, .1), 1))) %>%
  count()
```

```
## # A tibble: 7 x 2
## # Groups:   pct [7]
##   pct           n
##   <fct>     <int>
## 1 (0,0.2]      51
## 2 (0.2,0.3]    77
## 3 (0.3,0.4]   115
## 4 (0.4,0.5]   209
## 5 (0.5,0.6]   225
## 6 (0.6,0.7]   197
## 7 (0.7,1]      82
```

```
df_results_neighbourhood %>%
  select(AKP_two_march, ABS_share_diff) %>%
  group_by(pct=cut(AKP_two_march, c(0, seq(.2, .7, .1), 1))) %>%
  summarize(mean(ABS_share_diff))
```

```
## # A tibble: 7 x 2
##   pct       `mean(ABS_share_diff)`
##   <fct>                      <dbl>
## 1 (0,0.2]                -0.0588
## 2 (0.2,0.3]              -0.0475
## 3 (0.3,0.4]              -0.0343
## 4 (0.4,0.5]              -0.0245
## 5 (0.5,0.6]              -0.0232
## 6 (0.6,0.7]              -0.0211
## 7 (0.7,1]                -0.00566
```

Plot

- cluster standards errors at the district level when estimating subgroup means

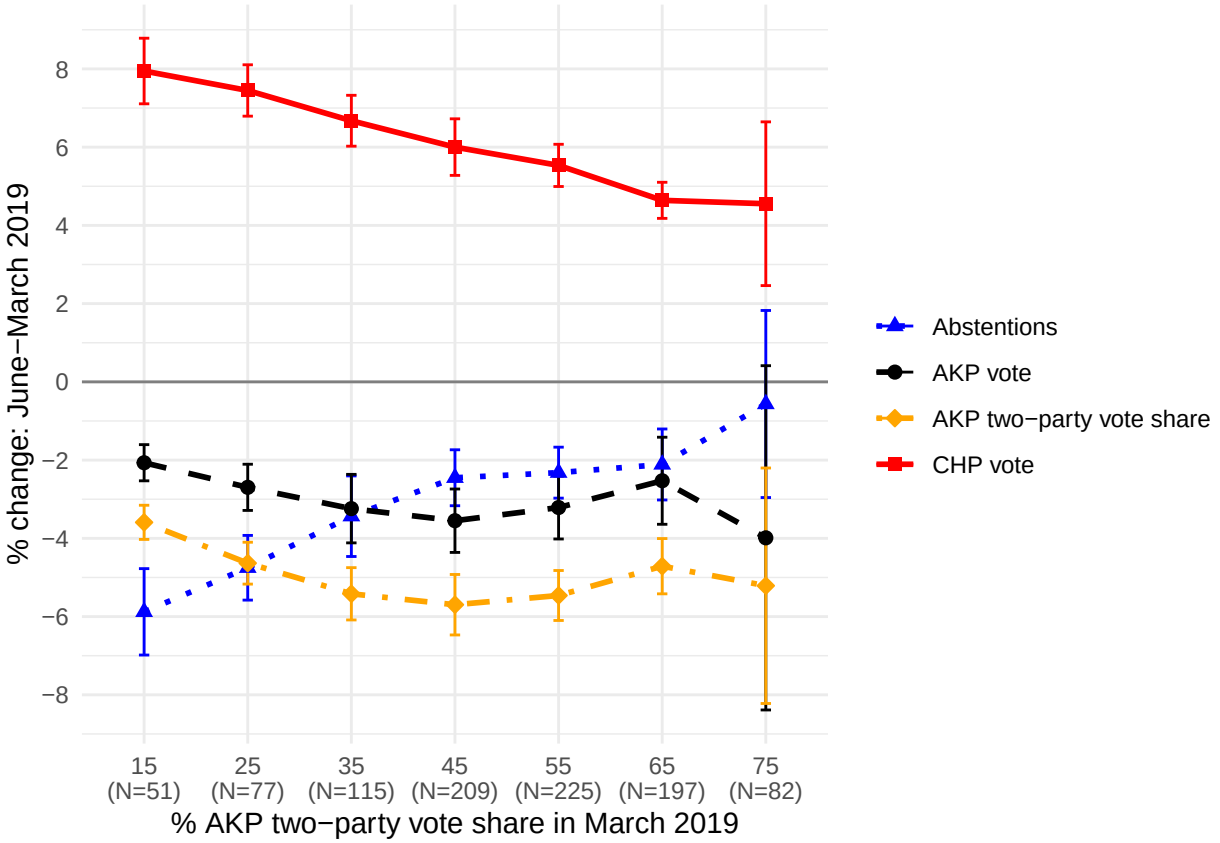
- Note: the confidence intervals obtained this way are more conservative than those obtained using hierarchical bootstrap (see below); so I am presenting the more conservative of the two

```
# RESHAPE THE DATA SO THAT IT IS IN A FORMAT THAT IS SUITABLE FOR PLOTTING
df_plot <- df_results_neighbourhood %>%
  select(district_id, AKP_two_march, AKP_share_diff, CHP_share_diff, ABS_share_diff, AKP_two_diff) %>%
  pivot_longer(cols = ends_with("_diff"), names_to = "outcome", values_to = "diff")

# CREATE X-AXIS SUBGROUPS/BINS AND ESTIMATE THEIR MEANS
df_plot_est_temp <- df_plot %>%
  group_by(pct=cut(AKP_two_march, c(0, seq(.2, .7, .1), 1)), group=as.factor(outcome)) %>%
  do(lm_robust(diff ~ 1, data = .) %>% tidy)

df_plot_est <- df_plot %>%
  group_by(pct=cut(AKP_two_march, c(0, seq(.2, .7, .1), 1)), group=as.factor(outcome)) %>%
  do(lm_robust(diff ~ 1, cluster=district_id, data = .) %>% tidy)


plot_group_labels <- c("Abstentions", "AKP vote", "AKP two-party vote share", "CHP vote")
df_plot_est %>%
  ggplot(aes(x=pct, y=estimate), col=group) +
  geom_hline(yintercept=0, col="grey50") +
  geom_line(aes(col=group, group=group, linetype = group), linewidth=1) +
  geom_errorbar(aes(ymin=conf.low, ymax=conf.high, col=group), width=0.1) +
  geom_point(aes(col=group, shape=group, fill=group), size=2) +
  scale_shape_manual(name  ="", values=c(17,19,23,15), labels=plot_group_labels) +
  scale_linetype_manual(name  ="", values=c("dotted", "dashed", "dotdash", "solid"), labels=plot_group_
  scale_color_manual(name  ="", values=c("blue", "black", "orange", "red"), labels=plot_group_labels) +
  scale_fill_manual(name  ="", values=c("blue", "black", "orange", "red"), labels=plot_group_labels) +
  theme_minimal() +
  scale_x_discrete(name = "% AKP two-party vote share in March 2019",
        labels=c("15\n(N=51)", "25\n(N=77)", "35\n(N=115)", "45\n(N=209)", "55\n(N=225)", "65\n(N=197)"
  scale_y_continuous(name = "% change: June-March 2019",
        labels=seq(-10, 10, 2),
        breaks=seq(-.1, .1, .02))
```




```
# SAVE THE PLOT
ggsave("figures/figure_2.png",  dpi = 1200, width = 10, height = 6, units = "in")
ggsave("figures/figure_2.pdf", dpi = 1200, width = 10, height = 6, units = "in")
ggsave("figures/figure_2.eps", dpi = 1200, width = 10, height = 6, units = "in")
```

Alternative plot: Bootstrap the CIs for Figure 2

- Hierarchical bootstrap (at the level of neighborhoods): mimic the data-generating process (and any dependence within neighborhoods):
  - First, draw neighborhoods (with replacement)
  - Second, from each drawn neighborhood, draw ballot boxes from that neighborhood (with replacement)
  - Third, compute the statistic of interest
  - Repeats boot_rep times

```
df_boot <- df_results

# NUMBER OF BOOTSTRAP DRAWS
boot_rep <- 10000

# NUMBER OF NEIGHBORHOODS
neighborhood_index <- unique(df_results$neighborhood_id)
length(neighborhood_index)

# A DF FOR THE BOOTSTRAPPED STATISTIC
df_means <- c()
```

```r
for (i in 1:boot_rep) {
  # SAMPLE THE NEIGHBORHOOD IDS
  boot_index_neighborhood <- sample(neighborhood_index, replace = TRUE)

  # THIS WILL BE THE DF OF BOOTSTRAPPED NEIGHBORHOODS
  df_neighborhood_boot <- c()
  for (j in 1:length(boot_index_neighborhood)) {

    # FILTER OUT NEIGHBORHOOD j
    df_neighborhood_j <- df_boot %>% filter(neighborhood_id == boot_index_neighborhood[j])

    # SAMPLE BOXES FROM NEIGHBORHOOD j
    box_length <- nrow(df_neighborhood_j)
    boot_index_box <- sample.int(box_length, replace = TRUE)
    df_boot_j <- df_neighborhood_j[boot_index_box, ]

    # STACK UP IN INTO THE BOOTSTRAPPED DF
    df_neighborhood_boot <- bind_rows(df_neighborhood_boot, df_boot_j)
  }

  # CONSTRUCT THE TEST STATISTIC
  df_results_neighbourhood_mean_boot <- df_neighborhood_boot %>%
    group_by(district_id, district_name, neighborhood_id, neighborhood_name) %>%
    summarize(AKP_march=sum(AKP_march), CHP_march=sum(CHP_march), ABS_march=sum(ABS_march), AKP_june=su
    ungroup() %>%
  mutate(
    REG_march = AKP_march + CHP_march + ABS_march,
    AKP_share_march = AKP_march/REG_march,
    CHP_share_march = CHP_march/REG_march,
    ABS_share_march = ABS_march/REG_march,
    REG_june = AKP_june + CHP_june + ABS_june,
    AKP_share_june = AKP_june/REG_june,
    CHP_share_june = CHP_june/REG_june,
    ABS_share_june = ABS_june/REG_june,
    AKP_share_diff = AKP_share_june - AKP_share_march,
    CHP_share_diff = CHP_share_june - CHP_share_march,
    ABS_share_diff = ABS_share_june - ABS_share_march,
    AKP_two_march = AKP_march/(AKP_march + CHP_march),
    AKP_two_june = AKP_june/(AKP_june + CHP_june),
    AKP_two_diff = AKP_two_june - AKP_two_march
  ) %>%
  select(AKP_two_march, AKP_share_diff, CHP_share_diff, ABS_share_diff, AKP_two_diff) %>%
  pivot_longer(!AKP_two_march, names_to = "outcome", values_to = "diff") %>%
  group_by(pct=cut(AKP_two_march, c(0, seq(.2, .7, .1), 1)), outcome=as.factor(outcome)) %>%
  summarize(mean = mean(diff, na.rm=T))

  #SAVE IT
  df_means <- bind_cols(df_means,  df_results_neighbourhood_mean_boot$mean)
}


# COMPUTE THE 95% CI
conf.low <- apply(df_means, 1, quantile, prob=.025)
```

```
conf.high <- apply(df_means, 1, quantile, prob=.975)
```

Construct the plot (not in the paper)

```
# RESHAPE THE DATA SO THAT IT IS IN A FORMAT THAT IS SUITABLE FOR PLOTTING
df_plot <- df_results_neighbourhood %>%
  select(AKP_two_march, AKP_share_diff, CHP_share_diff, ABS_share_diff, AKP_two_diff) %>%
  pivot_longer(!AKP_two_march, names_to = "outcome", values_to = "diff")

# CREATE X-AXIS SUBGROUPS/BINS AND ESTIMATE THEIR MEANS
df_plot <- df_plot %>%
  group_by(pct=cut(AKP_two_march, c(0, seq(.2, .7, .1), 1)), group=as.factor(outcome)) %>%
  summarize(
        estimate = mean(diff, na.rm=T))

df_plot <- df_plot %>% ungroup() %>%
  mutate(conf.low) %>%
  mutate(conf.high)

group_labels <- c("abstain/registered", "AKP vote/registered", "AKP two-party vote share", "CHP vote/reg
df_plot %>%
  ggplot(aes(x=pct, y=estimate), col=group) +
  geom_hline(yintercept=0, col="grey50") +
  geom_point(aes(col=group, shape=group), size=2) +
  geom_line(aes(col=group, group=group, linetype = group),  size=1) +
  scale_shape_manual(name  ="", values=c(18,16,15,17), labels=group_labels) +
  scale_linetype_manual(name  ="", values=c("dotdash", "dashed", "solid", "dotted"), labels=group_labels
  scale_color_manual(name  ="", values=c("blue", "black", "orange", "red"), labels=group_labels) +
  geom_errorbar(aes(ymin=conf.low, ymax=conf.high, col=group), width=0.1) +
  theme_minimal() +
  scale_x_discrete(name = "% AKP two-party vote share in March 2019",
        labels=c("15\n(N=51)", "25\n(N=77)", "35\n(N=115)", "45\n(N=209)", "55\n(N=225)", "65\n(N=197)"
  scale_y_continuous(name = "% change: June-March 2019",
        labels=seq(-10, 10, 2),
        breaks=seq(-.1, .1, .02))
```

Table 1

- ELECTION RESULTS: NEIGHBORHOOD-LEVEL SHIFTS IN THE VOTE FOR THE AKP, FOR THE CHP, AND ABSTENTION BETWEEN MARCH AND JUNE 2019

Load the data

```
rm(list = ls(all = TRUE))
load("data/df_2019istanbul_results.RData")
```

Aggregate results at the neighborhood level

```
# CHECK ON THE SHARE OF THIRD-PARTY VOTERS
df_results %>%
  mutate(across(AKP_march:THIRD_june, sum)) %>%
```

```
  slice(1) %>%
  mutate(
    THIRD_share_march = THIRD_march/(AKP_march + CHP_march + ABS_march + THIRD_march),
    THIRD_share_june = THIRD_june/(AKP_june + CHP_june + ABS_june + THIRD_june)) %>%
  select(starts_with("THIRD"))
```

```
## # A tibble: 1 x 4
##   THIRD_march THIRD_june THIRD_share_march THIRD_share_june
##         <dbl>      <dbl>             <dbl>            <dbl>
## 1      223657      68318            0.0212          0.00647
```

```
# TREAT THIRD-PARTY VOTERS AS ABSTAINERS
df_results <- df_results %>%
  mutate(
    ABS_march = ABS_march + THIRD_march,
    ABS_june = ABS_june + THIRD_june
    ) %>%
  select(-THIRD_march, -THIRD_june)

# AGGREGATE THE RESULTS AT THE NEIGHBORHOOD LEVEL
df_results_neighbourhood <- df_results %>%
  group_by(district_id, district_name, neighborhood_id, neighborhood_name) %>%
  summarize(AKP_march=sum(AKP_march), CHP_march=sum(CHP_march), ABS_march=sum(ABS_march), AKP_june=sum(A
```

```
## `summarise()` has grouped output by 'district_id', 'district_name',
## 'neighborhood_id'. You can override using the `.groups` argument.
```

SIMULATE THE NULL HYPOTHESIS

- Take the March fraction of citizens that voted for the AKP, the CHP, and abstained as the respective outcome probabilities under the null hypothesis
- Draw a large number of draws from the multinomial distribution for each neighborhood
- This simulation accounts for:
  - differences in neighborhood size: the same percentage shift in an outcome is less likely to occur by chance in neighborhoods with a larger number of registered voters
  - the fact that a negative correlation between shifts in any two outcomes arises mechanically because the number of registered voters is constant across the two elections

```
# A FUNCTION THAT RETURNS THE PERCENTILE OF THE VALUE x IN THE EMPIRICAL DISTRIBUTION OF sample
centile <- function(sample, x) {
              ecdf(sample)(x)
}

# DATA LENGTH
neighborhoods_length <- nrow(df_results_neighbourhood)

# NUMBER OF DRAWS PER NEIGHBORHOOD
rep <- 10000

# LISTS TO SAVE SIMS INTO
vote_centiles_list <- vector("list", neighborhoods_length)

# LOOP OVER ALL NEIGHBORHOODS
for (i in 1:neighborhoods_length) {
```

```
  # ISOLATE MARCH AND JUNE OUTCOMES FOR NEIGHBORHOOD i
  df_neighborhood_i <- df_results_neighbourhood[i,]
  march_i <- c(df_neighborhood_i$AKP_march, df_neighborhood_i$CHP_march, df_neighborhood_i$ABS_march)
  june_i <- c(df_neighborhood_i$AKP_june, df_neighborhood_i$CHP_june, df_neighborhood_i$ABS_june)

  # THE NULL/MARCH PROBABILITIES
  P_march_i <- march_i/sum(march_i)
  # rep DRAWS FROM THE NULL DISTRIBUTION
  NULL_draw_i <- rmultinom(n=rep, size=sum(june_i), prob=P_march_i)

  # THE QUANTILE OF THE OBSERVED VALUE: HOW MUCH OF AN OUTLIER IS THE JUNE OUTCOME BY THE STANDARD OF T
  AKP_centile <- centile(NULL_draw_i[1,], june_i[1])
  CHP_centile <- centile(NULL_draw_i[2,], june_i[2])
  ABS_centile <- centile(NULL_draw_i[3,], june_i[3])

  # SAVE THE CENTILES FOR NEIGHBORHOOD i
  vote_centiles_list[[i]] <- c(AKP_centile, CHP_centile, ABS_centile)

  }
```

Which June election outcomes depart from March ones too much to have occurred solely due to the idiosyncratic randomness inherent in elections?

- identify all neighborhoods where at least one of the three June outcomes falls below the 2.5th or above the 97.5th percentile of the simulated draws

```
# MATRIX WITH THE CENTILE COLUMNS: AKP, CHP, ABS
centiles <- matrix(unlist(vote_centiles_list), ncol=3, byrow=TRUE)

# CLASSIFY INTO "above", "below" (the 2.5th or above the 97.5th percentile of the simulated draws) or "
outside <- ifelse(centiles<.025, "below", centiles)
outside <- ifelse(centiles>.975, "above", outside)
outside <- ifelse(!(centiles<.025 | centiles>.975) , "neither", outside)

apply(outside, 2, table)
```

```
## [[1]]
##
##   above   below neither
##       7     763     186
##
## [[2]]
##
##   above neither
##     888      68
##
## [[3]]
##
##   above   below neither
##      40     720     196
```

```
apply(outside, 2, function(x) prop.table(table(x)))
```

```
## [[1]]
## x
```

```
##       above       below     neither
## 0.007322176 0.798117155 0.194560669
##
## [[2]]
## x
##      above    neither
## 0.92887029 0.07112971
##
## [[3]]
## x
##     above     below   neither
## 0.0418410 0.7531381 0.2050209
```

Construct Table 1

```
# COMBINE INTO A VECTOR THAT LISTS WHETHER AKP VOTE, CHP VOTE OR ABSTENTION IS OUTSIDE THE CI
combine <- apply(outside, 1, function(x) paste(x, collapse=", "))
length(combine)
```

```
## [1] 956
```

```
sort(table(combine))
```

```
## combine
##   neither, neither, below   neither, neither, above       above, above, below
##                         1                         2                         7
##   below, neither, neither     below, neither, above   neither, above, neither
##                        12                        18                        18
##       below, above, above neither, neither, neither     neither, above, below
##                        20                        35                       130
##     below, above, neither       below, above, below
##                       131                       582
```

```
round(100*sort(prop.table(table(combine))), 2)
```

```
## combine
##   neither, neither, below   neither, neither, above       above, above, below
##                      0.10                      0.21                      0.73
##   below, neither, neither     below, neither, above   neither, above, neither
##                      1.26                      1.88                      1.88
##       below, above, above neither, neither, neither     neither, above, below
##                      2.09                      3.66                     13.60
##     below, above, neither       below, above, below
##                     13.70                     60.88
```

```
table_1 <- combine %>%
  as_tibble() %>%
  group_by(value) %>%
  summarise(N=n()) %>%
  mutate(Percentage = round(100*N/sum(N), 2)) %>%
  arrange(desc(Percentage)) %>%
  mutate(`Cumulative Percentage` = cumsum(Percentage))

table_1
```

```
## # A tibble: 11 x 4
```

```
##    value                          N Percentage `Cumulative Percentage`
##    <chr>                      <int>      <dbl>                   <dbl>
##  1 below, above, below          582      60.9                     60.9
##  2 below, above, neither        131      13.7                     74.6
##  3 neither, above, below        130      13.6                     88.2
##  4 neither, neither, neither     35       3.66                    91.8
##  5 below, above, above           20       2.09                    93.9
##  6 below, neither, above         18       1.88                    95.8
##  7 neither, above, neither       18       1.88                    97.7
##  8 below, neither, neither       12       1.26                    99.0
##  9 above, above, below            7       0.73                    99.7
## 10 neither, neither, above        2       0.21                    99.9
## 11 neither, neither, below        1       0.1                    100.
```

```
table_1 %>%
  slice(c(1:4, 6))
```

```
## # A tibble: 5 x 4
##   value                          N Percentage `Cumulative Percentage`
##   <chr>                      <int>      <dbl>                   <dbl>
## 1 below, above, below          582      60.9                     60.9
## 2 below, above, neither        131      13.7                     74.6
## 3 neither, above, below        130      13.6                     88.2
## 4 neither, neither, neither     35       3.66                    91.8
## 5 below, neither, above         18       1.88                    95.8
```

```
table_1 %>%
  slice(-c(1:4, 6))
```

```
## # A tibble: 6 x 4
##   value                          N Percentage `Cumulative Percentage`
##   <chr>                      <int>      <dbl>                   <dbl>
## 1 below, above, above           20       2.09                    93.9
## 2 neither, above, neither       18       1.88                    97.7
## 3 below, neither, neither       12       1.26                    99.0
## 4 above, above, below            7       0.73                    99.7
## 5 neither, neither, above        2       0.21                    99.9
## 6 neither, neither, below        1       0.1                    100.
```

```
table_1 %>%
  slice(-c(1:4, 6)) %>%
  summarize(sum(N), sum(Percentage))
```

```
## # A tibble: 1 x 2
##   `sum(N)` `sum(Percentage)`
##      <int>             <dbl>
## 1       60              6.27
```