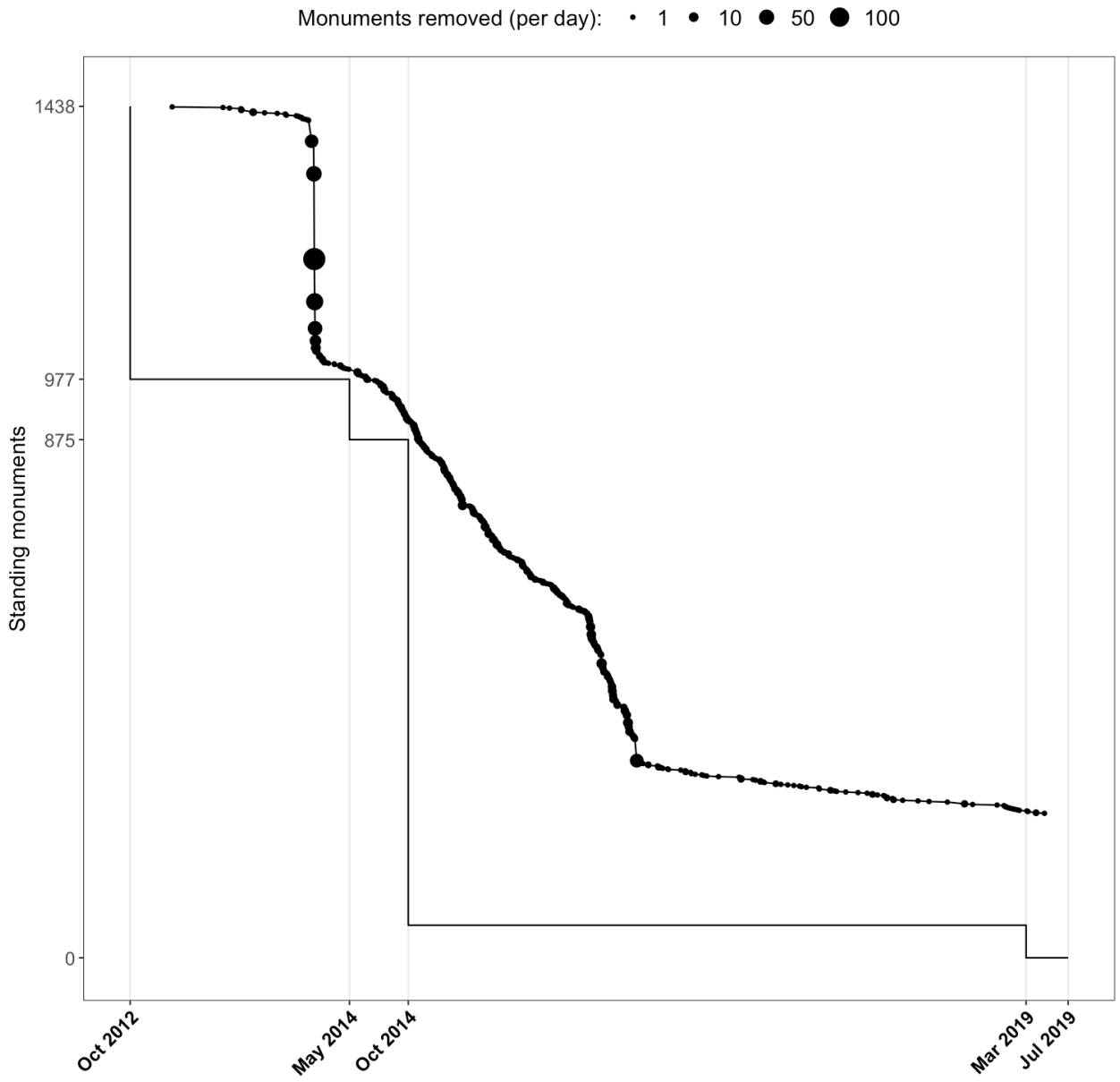
The results below are generated from an R script.

```
# set R to source file location

start <- Sys.time()
library(pacman)
p_load("sf", "lfe", "lubridate", "data.table", "stargazer",
       "RColorBrewer", "dplyr", "zoo", "gsynth", "randomForest", "verification", "tm", "stm",  "xtable",
       "stringi", "stringr", "Formula")
set.seed(1)
```

```
#######################################
# Define new functions
#######################################

# Cutoffs for p-values
cutoff <- function(x, p){
x <- roundr(x, 1)
y <- c("^{***}", "^{**}", "^{*}", "")
p <- cut(p, breaks = c(0, 0.001, 0.01, 0.05, Inf),
         include.lowest = TRUE, labels = y)
paste(x, p, sep = "")
}

# regression output processing
coef_out <- function(z, name = NULL, below = TRUE, dollar = FALSE, round = 1){
  w <- grep(name, names(coefficients(z)))
  names <- names(coefficients(z))[w]
  y <- matrix(coefficients(summary(z))[w, ], nrow = length(w))
  b <- roundr(y[, 1], round)
  p <- y[, 4]
  p <- ifelse(p < 0.001, "^{***}", ifelse(p < 0.01, "^{**}", ifelse(p < 0.05, "^{*}", "")))
  se <- paste("(", roundr(y[,2],  round), ")", sep="")
  s <- ifelse(dollar, "$", "")
  if(!below) return(data.table(var = names, b = paste(s, b, "~", se, p, s, sep = "")))
  if(below) return(rbind(b = paste0(s, b, p, s), se = se) %>% c()
                   %>% data.table() %>% setnames("b")
                   %>% data.table(var = rep(names, each = 2), type = rep(c("Coef", "SE"),  nrow(.)/2)))
}

# nicer rounding
roundr <- function(x, n = 2) {
  f <- function(x, n) sprintf(paste("%.", n, "f", sep=""), round(x, n))
  if (is.null(dim(x))) out <- f(x, n)
  if (!is.null(dim(x))) out <- apply(x, 2, f, n = n)
  out
}

# Some functions produce a lot cat output, supress it
quiet <- function (..., messages = FALSE, cat = FALSE)
{
  if (!cat) {
    sink(tempfile())
    on.exit(sink())
```

```
  }
  out <- if (messages)
    eval(...)
  else suppressMessages(eval(...))
  out
}

################################################
# Figure 1: Monument removals in time
################################################
lenin <- readRDS("statues.RDS")
# Select post-soviet statues and remove Donbass
lenin <- lenin[cycle > 0 & !region %in% c("Donetskaya", "Luhanskaya")]
#  Select statues where exact removal dates are available
Lenin <- lenin%>%
  group_by(date_ymd)%>%
  summarise(r = n())%>%ungroup()%>%
  mutate(y = sum(r) - cumsum(r)) %>% na.omit() %>% data.table()
Lenin <- Lenin[date_ymd != ymd("2019-12-31")]

  e <- as.Date(c("2012-10-28", "2014-05-25", "2014-10-26", "2019-03-31", "2019-07-19"))
  # Breaks at each election
  y <- c(nrow(lenin), nrow(lenin) - cumsum(lapply(list(1, 2, c(3, 4), c(5, 6)),
                      function(x) sum(lenin$cycle%in%x))%>%unlist()))
  g1 <- ggplot(Lenin, aes(x = as.Date(date_ymd), y = y)) +
    geom_point(aes(size = r))  + geom_line() + theme_bw() +
    theme(legend.position="top",
          legend.direction = "horizontal",
          legend.text=element_text(size = 14),
          legend.title = element_text(size = 14),
          panel.grid.major.y = element_blank(),
          panel.grid.minor = element_blank(),
          axis.text=element_text(size = 12),
          axis.title.y = element_text(size = 14),
          axis.text.x=element_text(size = 12, face = "bold", color = "black", angle = 45, hjust = 1),
          axis.title=element_text(size = 12)) + xlab("") +
    scale_size_continuous(breaks = c(1, 10, 50, 100)) +
    labs(size = "Monuments removed (per day): ") +
    scale_x_date(breaks = e, labels = as.yearmon(e)) +
    scale_y_continuous(breaks = y[-4], limits = c(0, 1450)) +
    ylab("Standing monuments")
  # Add the step function
  s <- data.table(x = c(as.Date("2012-10-28"), e), y = c(y, last(y)))
  g1 + geom_step(data = s, aes(x = x, y = y))
```



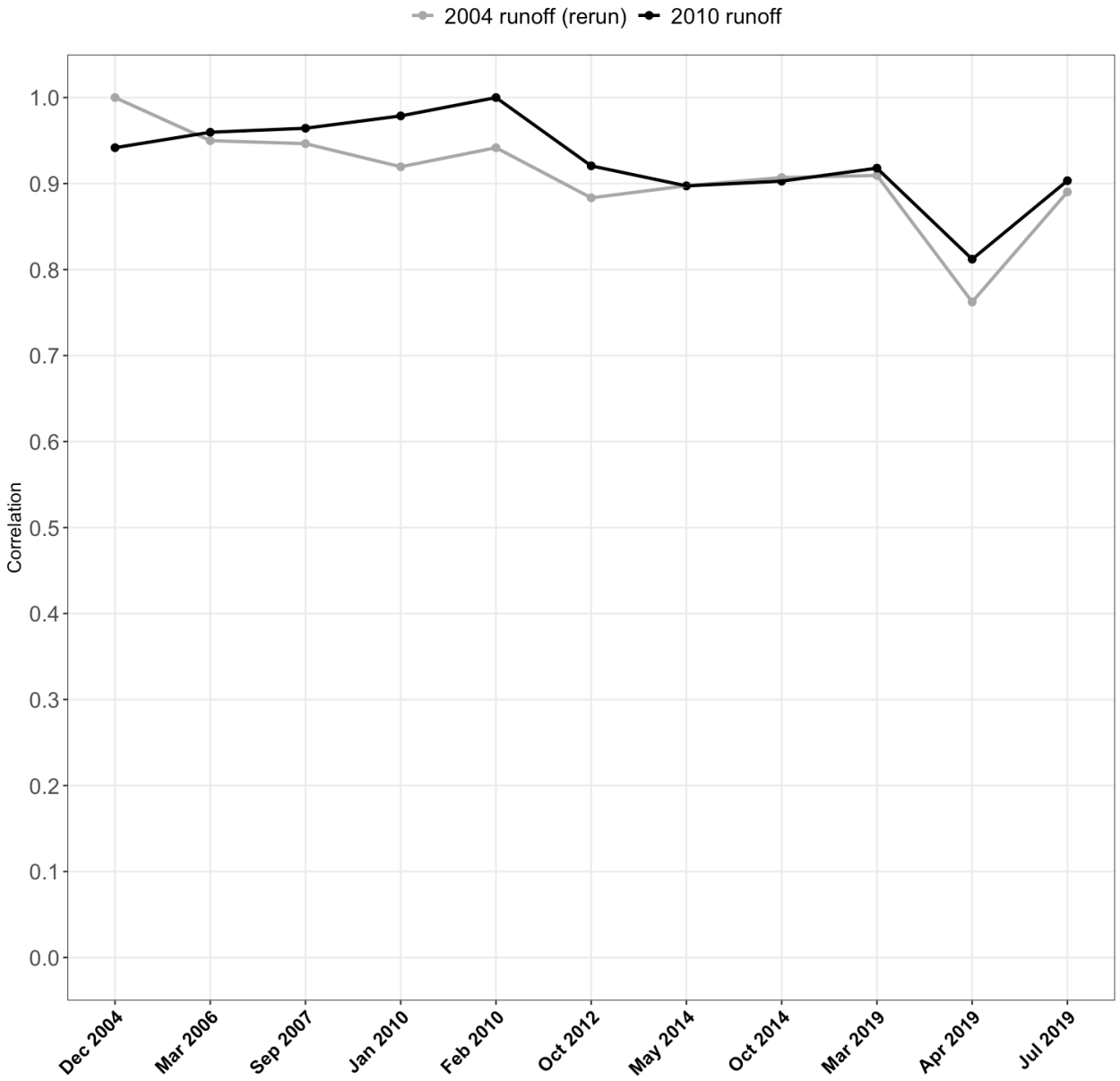

```
#####################################################
# Figure 2: Locations of the Soviet monuments
#####################################################
# Load maps
ukr <- readRDS("UKR_adm1.RDS") %>% st_as_sf()
# Re-load statues
lenin <- readRDS("statues.RDS")
# Add geography
lenin <- st_as_sf(subset(lenin, !is.na(lon)),
                  coords = c("lon", "lat"),
                  crs = st_crs(ukr),
                  agr = "constant")
names <- c("Removed before Euromaidan", "Removed during/after Euromaidan", "Standing / NA")
lenin$removed[lenin$cycle==0] <- names[1]
```

```r
lenin$removed[lenin$cycle > 0 & lenin$cycle < 6] <- names[2]
lenin$removed[lenin$cycle == 6 | is.na(lenin$cycle)] <- names[3]
lenin$removed <- factor(lenin$removed, levels = names)
cols <- c(brewer.pal(9, "Blues")[c(5, 9)], "#FF8000")
ggplot() + geom_sf(data = ukr, fill = "white", col = "black", lwd = 0.3, alpha = 0.1) +
  coord_sf(crs = st_crs(ukr), datum = NA) +
  geom_sf(data = lenin, aes(shape = removed, col = removed), alpha = .75, cex = 1.5) +
  coord_sf(crs = st_crs(lenin), datum = NA) +
  theme_minimal() +
  scale_color_manual(name = "", values = c("black", "grey", "black")) +
  scale_shape_manual(name = "", values = c(2, 17, 4)) +
  theme(legend.position="top", legend.margin=ggplot2::margin(b=-20), legend.box.margin=ggplot2::margin(0
  guides(shape = guide_legend(override.aes = list(size=4)))
```

```
## Coordinate system already present.  Adding new coordinate system, which will replace the
existing one.
```

```
#####################################################
# Data and main results
#####################################################

# Data on elections and removals:
d <- readRDS("main_data.RDS")

# Figure 1 (Appendix A): Validating measures
d[d[election == ymd("2004-12-25")], on = "name", y_2004 := i.pro_russian]
d[d[election == ymd("2010-02-07")], on = "name", y_2010 := i.pro_russian]
D <- d[, lapply(.SD, function(x) cor(pro_russian, x, use = "complete")),
       by = election, .SDcols = c("y_2004", "y_2010")] %>% melt(id.vars = c("election"))

ggplot(data = D, aes(x = as.factor(election) %>% as.numeric(), y=value, color = variable)) +
```

```
  geom_point(size = 2) + geom_line(size = 1) +
  theme_bw() +
  ylab("Correlation") + xlab("") +
  scale_x_continuous(labels = d$election %>% unique() %>% sort() %>% as.yearmon(), breaks = 1:11) +
  scale_color_manual(labels = c("2004 runoff (rerun)", "2010 runoff"), values = c("dark grey", "black")) +
  scale_y_continuous(breaks = seq(0, 1, by = 0.1), lim = c(0, 1)) +
  theme(
  legend.position="top",
  legend.title = element_blank(),
  legend.text=element_text(size = 14),
  panel.grid.minor = element_blank(),
  axis.text=element_text(size = 14),
  axis.text.x=element_text(size = 12, face = "bold", color = "black", angle = 45, hjust = 1),
  axis.title=element_text(size = 12))
```



```r
# Remove 2019 presidential runoff
d <- d[election != ymd("2019-04-21")]
d[, cycle := which(election == unique(d$election)%>%ymd()%>%sort()) %>% as.numeric(), by = election]
d <- d[order(name, election)]
d[, time := as.numeric(cycle)]

# Select precincts that had any statues prior to Leninopad
d <- d[name %in% d[n > 0 & election == ymd("2012-10-28"), name]]
# Number of statues at time t
d[, n := ifelse(is.na(n), max(n, na.rm = TRUE), n), by = name]
# Number of statues removed by time t
d[, X := max(n) - n, by = name]

# List of models (specifications)
m <- Formula(y ~ X | name + election | 0 | name + oblast)
models <- list(
  m,
  update(m, . ~ . | . + time:oblast),
  update(m, . ~ . + time:(log(dist_to_kiev) + lon*lat + log(km_road)) | . + time:oblast + time:size),
  update(m, . ~ . | . + name:time)
)

# Use this to replicate estimation for each outcome
outcomes <- function(D, m){
  list(
    felm(update(m, turnout ~ .), data = D, weights = D$voters),
    felm(update(m, ru_turnout ~ .), data = D, weights = D$voters))
}

############################################################
# Table 1: Estimates from multi-period DiD regressions
############################################################

# This vector collects the estimates
z <- lapply(models, outcomes, D = d)

stargazer(z[[1]][[1]], z[[2]][[1]], z[[3]][[1]], z[[4]][[1]],
          style="apsr",
          label = "tab:main_votes",
          omit.table.layout = "n", omit.stat=c("rsq", "f", "ser"), omit=c("Constant"),
          float = T, digits = 1,
          title = "Diff-in-diff estimates combining all elections.",
          add.lines = list(
            c("Precincts FE", rep("\\checkmark", 4)),
            c("Election FE", "\\checkmark", rep("", 3)),
            c("Oblast FE $\\times$ time", "", rep("\\checkmark", 3)),
            c("Covariates $\\times$ time", c("", "", "\\checkmark", "\\checkmark")),
            c("Precincts FE $\\times$ time", c("", "", "", "\\checkmark"))),
          covariate.labels = c("Effect on overall turnout"),
          dep.var.labels.include = FALSE,
          no.space = TRUE,
          keep = "X",
          star.cutoffs = c(.05, 0.01, 0.001))
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harva
## % Date and time: Thu, Sep 02, 2021 - 12:19:16
## \begin{table}[!htbp] \centering
##   \caption{Diff-in-diff estimates combining all elections.}
##   \label{tab:main_votes}
## \begin{tabular}{@{\extracolsep{5pt}}lcccc}
## \\[-1.8ex]\hline \\[-1.8ex]
## \\[-1.8ex] & (1) & (2) & (3) & (4)\\
## \hline \\[-1.8ex]
##  Effect on overall turnout & 3.6$^{***}$ & 3.4$^{***}$ & 3.3$^{***}$ & 4.3$^{***}$ \\
##   & (0.6) & (0.6) & (0.6) & (0.8) \\
##  Precincts FE & \checkmark & \checkmark & \checkmark & \checkmark \\
## Election FE & \checkmark &  &  &  \\
## Oblast FE $\times$ time &  & \checkmark & \checkmark & \checkmark \\
## Covariates $\times$ time &  &  & \checkmark & \checkmark \\
## Precincts FE $\times$ time &  &  &  & \checkmark \\
## N & 11,860 & 11,860 & 11,860 & 11,860 \\
## Adjusted R$^{2}$ & 0.8 & 0.8 & 0.8 & 0.8 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}

stargazer(z[[1]][[2]], z[[2]][[2]], z[[3]][[2]], z[[4]][[2]],
          style="apsr",
          label = "tab:main_turnout",
          omit.table.layout = "n", omit.stat=c("rsq", "f", "ser"), omit=c("Constant"),
          float = T, digits = 1,
          title = "Diff-in-diff estimates combining all elections.",
          add.lines = list(
            c("Precincts FE", rep("\\checkmark", 4)),
            c("Election FE", "\\checkmark", rep("", 3)),
            c("Oblast-election FE", "", rep("\\checkmark", 3)),
            c("Covariates", c("", "", "\\checkmark", "\\checkmark")),
            c("Time trend", c("", "", "", "\\checkmark"))),
          covariate.labels = c("Effect on pro-Soviet turnout"),
          dep.var.labels.include = FALSE,
          no.space = TRUE,
          keep = "X",
          star.cutoffs = c(.05, 0.01, 0.001)
)

##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harva
## % Date and time: Thu, Sep 02, 2021 - 12:19:16
## \begin{table}[!htbp] \centering
##   \caption{Diff-in-diff estimates combining all elections.}
##   \label{tab:main_turnout}
## \begin{tabular}{@{\extracolsep{5pt}}lcccc}
## \\[-1.8ex]\hline \\[-1.8ex]
## \\[-1.8ex] & (1) & (2) & (3) & (4)\\
## \hline \\[-1.8ex]
##  Effect on pro-Soviet turnout & 2.4$^{***}$ & 1.6$^{***}$ & 1.5$^{***}$ & 1.8$^{***}$ \\
##   & (0.5) & (0.3) & (0.3) & (0.4) \\
```

```
##  Precincts FE & \checkmark & \checkmark & \checkmark & \checkmark \\
## Election FE & \checkmark &  &  &  \\
## Oblast-election FE &  & \checkmark & \checkmark & \checkmark \\
## Covariates &  &  & \checkmark & \checkmark \\
## Time trend &  &  &  & \checkmark \\
## N & 11,860 & 11,860 & 11,860 & 11,860 \\
## Adjusted R$^{2}$ & 0.9 & 0.9 & 0.9 & 0.9 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

```
## Time as factor (Table B.2. Appendix B.1):
d[, time := election]
z <- lapply(models, outcomes, D = d)

stargazer(z[[1]][[1]], z[[2]][[1]], z[[3]][[1]], z[[4]][[1]],
          style="apsr",
          label = "tab:main_votes_factor",
          omit.table.layout = "n", omit.stat=c("rsq", "f", "ser"), omit=c("Constant"),
          float = T, digits = 1,
          title = "Diff-in-diff estimates combining all elections.",
          add.lines = list(
            c("Precincts FE", rep("\\checkmark", 4)),
            c("Election FE", "\\checkmark", rep("", 3)),
            c("Oblast FE $\\times$ time", "", rep("\\checkmark", 3)),
            c("Covariates $\\times$ time", c("", "", "\\checkmark", "\\checkmark")),
            c("Precincts FE $\\times$ time", c("", "", "", "\\checkmark"))),
          covariate.labels = c("Effect on turnout"),
          dep.var.labels.include = FALSE,
          no.space = TRUE,
          keep = "X",
          star.cutoffs = c(.05, 0.01, 0.001))
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harva
## % Date and time: Thu, Sep 02, 2021 - 12:19:17
## \begin{table}[!htbp] \centering
##   \caption{Diff-in-diff estimates combining all elections.}
##   \label{tab:main_votes_factor}
## \begin{tabular}{@{\extracolsep{5pt}}lcccc}
## \\[-1.8ex]\hline \\[-1.8ex]
## \\[-1.8ex] & (1) & (2) & (3) & (4)\\
## \hline \\[-1.8ex]
##  Effect on turnout & 3.6$^{***}$ & 3.5$^{***}$ & 3.4$^{***}$ & 4.5$^{***}$ \\
##   & (0.6) & (0.6) & (0.7) & (0.9) \\
##  Precincts FE & \checkmark & \checkmark & \checkmark & \checkmark \\
## Election FE & \checkmark &  &  &  \\
## Oblast FE $\times$ time &  & \checkmark & \checkmark & \checkmark \\
## Covariates $\times$ time &  &  & \checkmark & \checkmark \\
## Precincts FE $\times$ time &  &  &  & \checkmark \\
## N & 11,860 & 11,860 & 11,860 & 11,860 \\
## Adjusted R$^{2}$ & 0.8 & 0.8 & 0.8 & 0.8 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

```
stargazer(z[[1]][[2]], z[[2]][[2]], z[[3]][[2]], z[[4]][[2]],
          style="apsr",
          label = "tab:main_turnout_factor",
          omit.table.layout = "n", omit.stat=c("rsq", "f", "ser"), omit=c("Constant"),
          float = T, digits = 1,
          title = "Diff-in-diff estimates combining all elections.",
          add.lines = list(
            c("Precincts FE", rep("\\checkmark", 4)),
            c("Election FE", "\\checkmark", rep("", 3)),
            c("Oblast-election FE", "", rep("\\checkmark", 3)),
            c("Covariates", c("", "", "\\checkmark", "\\checkmark")),
            c("Time trend", c("", "", "", "\\checkmark"))),
          covariate.labels = c("Effect on pro-Soviet turnout"),
          dep.var.labels.include = FALSE,
          no.space = TRUE,
          keep = "X",
          star.cutoffs = c(.05, 0.01, 0.001)
)

##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harva
## % Date and time: Thu, Sep 02, 2021 - 12:19:17
## \begin{table}[!htbp] \centering
##   \caption{Diff-in-diff estimates combining all elections.}
##   \label{tab:main_turnout_factor}
## \begin{tabular}{@{\extracolsep{5pt}}lcccc}
## \\[-1.8ex]\hline \\[-1.8ex]
## \\[-1.8ex] & (1) & (2) & (3) & (4)\\
## \hline \\[-1.8ex]
##  Effect on pro-Soviet turnout & 2.4$^{***}$ & 1.9$^{***}$ & 1.8$^{***}$ & 2.3$^{***}$ \\
##   & (0.5) & (0.4) & (0.4) & (0.5) \\
##  Precincts FE & \checkmark & \checkmark & \checkmark & \checkmark \\
## Election FE & \checkmark &  &  &  \\
## Oblast-election FE &  & \checkmark & \checkmark & \checkmark \\
## Covariates &  &  & \checkmark & \checkmark \\
## Time trend &  &  &  & \checkmark \\
## N & 11,860 & 11,860 & 11,860 & 11,860 \\
## Adjusted R$^{2}$ & 0.9 & 0.9 & 0.9 & 0.9 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}

## Recode "time" variable back to numeric, for later consistency
d[, time := as.numeric(cycle)]

## Two period specifications:

# Create two-period data
D_data <- function(elections){
  D <- copy(d)
  D <- D[election %in% ymd(elections)]
  D <- D[name %in% D[n > 0 & election %in% ymd(elections), name]]
  D <- D[order(name, cycle)]
  D[, n_start := max(n), by = name]
```

```
  D[, removed := 1*(n[election == ymd(elections[1])] > n[election == ymd(elections[2])]), by = name]
  D[, removed_any := 1*(n[1] > n[2]), by = name]
  D[, removed_all := 1*(n[2] == 0), by = name]
  D[, n_baseline := max(n, na.rm = TRUE), by = name]
  D[, removed_n := n_baseline - n, by = name]
  D[, removed_1 := 1*(n_baseline - n == 1), by = name]
  D[, removed_2 := 1*(n_baseline - n == 2), by = name]
  D[, post := 1*(election > ymd(elections[1])), by = name]
  D[, X := 1*(removed == 1 & post == 1)]
  D
}

x <- list(c("2012-10-28", "2014-05-25"),
          c("2014-05-25", "2014-10-26"),
          c("2014-10-26", "2019-03-31"),
          c("2014-10-26", "2019-07-21"))

out_each <- lapply(1:3, function(z) lapply(x, function(x) outcomes(D_data(x), m = models[[z]])))

# Full table (except the first election) for appendix and for paper for model 3
tab_out <- function(j){
  tab <- lapply(1:2, function(i) lapply(out_each[[j]], function(z) coef_out(z[[i]], name = "X", dollar =
  tab <- tab[ which(row.names(tab) %in% "b"),]
  # Which election is compared?
  e <- lapply(x, function(z) paste(as.yearmon(z), collapse = " - ")) %>% unlist()
  # Precincts per analysis
  n <- lapply(out_each[[j]], function(x) length(unique(x[[1]]$fe$name)))%>%unlist()%>%prettyNum(big.mark
  data.frame(e, tab, n)
}

########################################
# Table 2: Two-period DiD Regressions
########################################
tab_out(3) %>% kable()
```

	e	X1	X2	n
b	Oct 2012 - May 2014	$1.6~(0.5)^{**}$	$1.7~(0.4)^{***}$	1,296
b.1	May 2014 - Oct 2014	$-0.5~(0.4)$	$0.1~(0.2)$	887
b.2	Oct 2014 - Mar 2019	$-1.3~(0.7)$	$-0.6~(0.9)$	792
b.3	Oct 2014 - Jul 2019	$-0.2~(0.7)$	$-0.8~(0.8)$	792

```
########################################
# Table B.3. Appendix B.2:
########################################
lapply(1:2, tab_out) %>% rbindlist() %>% kable()
```

e	X1	X2	n
Oct 2012 - May 2014	$6.4~(1.3)^{***}$	$4.4~(1.0)^{***}$	1,296
May 2014 - Oct 2014	$-0.8~(0.6)$	$0.4~(0.5)$	887
Oct 2014 - Mar 2019	$-0.9~(0.9)$	$-0.9~(0.8)$	792
Oct 2014 - Jul 2019	$0.3~(1.2)$	$-1.1~(0.8)$	792
Oct 2012 - May 2014	$2.5~(0.4)^{***}$	$3.3~(0.4)^{***}$	1,296
May 2014 - Oct 2014	$-0.3~(0.4)$	$0.3~(0.2)$	887
Oct 2014 - Mar 2019	$-1.3~(0.8)$	$-0.9~(0.8)$	792
Oct 2014 - Jul 2019	$0.0~(0.7)$	$-0.9~(0.7)$	792

```
## Save the first pair as a separate dataset:
D <- D_data(c("2012-10-28", "2014-05-25"))

##########################################
# Table 3: Decomposition of the DiD effects
##########################################
dd_table <- function(felm_out){
  D[, Y := felm_out$fitted.values]
  c1 <- D[removed_any == 0 & post == 0, mean(Y)]
  c2 <- D[removed_any == 0 & post == 1, mean(Y)]
  t1 <- D[removed_any == 1 & post == 0, mean(Y)]
  t2 <- ((t1 + coefficients(felm_out)["X"] + (c2 - c1)))
  tab <- data.table(c1, c2, c.delta = c2-c1, t1, t2, t.delta = t2-t1, delta = (t2-t1) - (c2-c1))
  tab[, lapply(.SD, roundr, n = 1)][]
}

lapply(out_each[[3]][[1]], dd_table) %>%
  rbindlist() %>% kable()
```

c1	c2	c.delta	t1	t2	t.delta	delta
57.2	56.1	-1.1	57.5	58.0	0.5	1.6
31.0	9.9	-21.2	21.1	1.7	-19.5	1.7

```
#####################################
# Table B.5. Appendix B.4:
#####################################
outcomes_w <- function(D, m){
  list(
    felm(update(m, turnout ~ .), data = D),
    felm(update(m, ru_turnout ~ .), data = D))
}

out_each <- lapply(1:3, function(z) lapply(x, function(x) outcomes_w(D_data(x), m = models[[z]])))

lapply(1:3, tab_out) %>% rbindlist() %>% kable()
```

e	X1	X2	n
Oct 2012 - May 2014	$6.1~(1.2)^{***}$	$4.8~(0.9)^{***}$	1,296
May 2014 - Oct 2014	$-1.0~(0.6)$	$0.3~(0.5)$	887
Oct 2014 - Mar 2019	$-1.4~(1.1)$	$-1.1~(0.8)$	792
Oct 2014 - Jul 2019	$-0.3~(1.4)$	$-1.1~(0.7)$	792
Oct 2012 - May 2014	$2.7~(0.3)^{***}$	$3.8~(0.4)^{***}$	1,296
May 2014 - Oct 2014	$-0.6~(0.5)$	$0.4~(0.3)$	887
Oct 2014 - Mar 2019	$-1.0~(1.0)$	$-1.0~(0.8)$	792
Oct 2014 - Jul 2019	$0.2~(1.2)$	$-1.0~(0.7)$	792
Oct 2012 - May 2014	$1.7~(0.4)^{***}$	$1.7~(0.5)^{**}$	1,296
May 2014 - Oct 2014	$-0.6~(0.4)$	$0.2~(0.2)$	887
Oct 2014 - Mar 2019	$-0.8~(0.8)$	$-0.6~(0.8)$	792
Oct 2014 - Jul 2019	$0.1~(1.1)$	$-0.8~(0.7)$	792

```
#############################################################
#Table B.4. Appendix B.3 (Alternative definitions of treatment)
#############################################################
```

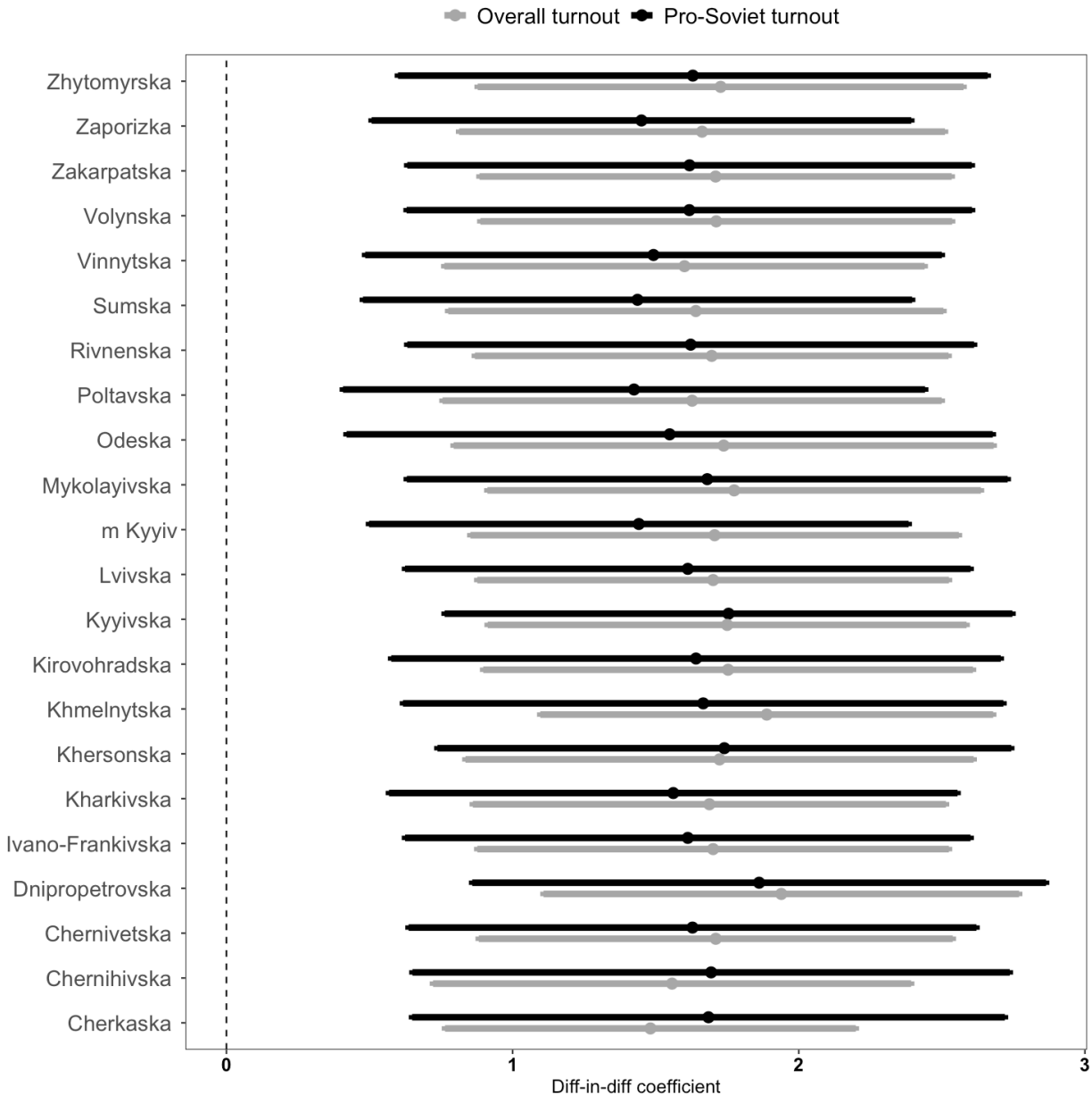
```
z <- list(
  outcomes(D, m = update(models[[3]], . ~ . - X + removed_all:post)),
  outcomes(D, m = update(models[[3]], . ~ . - X + removed_n:post)),
  outcomes(D, m = update(models[[3]], . ~ . - X + post:(removed_1 + removed_2))),
  outcomes(D, m = update(models[[3]], . ~ . - X + I(removed_n/n_start):post))
)

tab_out <- function(){
  vars <- c("Pro-Soviet", "Turnout")
  tab <- lapply(seq_along(z), function(j){
    lapply(1:2, function(i) data.table(coef_out(z[[j]][[i]], name = "removed", dollar = TRUE, below = TR
  rbindlist(tab)
}

tab <- tab_out()
tab[grepl("all", var), var := "All monuments removed"]
tab[grepl("removed_n:post", var), var := "Number of removed monuments"]
tab[grepl("removed_1", var), var := "One monument removed"]
tab[grepl("removed_2", var), var := "Two monuments removed"]
tab[grepl("removed_n/n_start", var), var := "Percentage of monuments removed"]
tab[type == "SE", var := ""][, type := NULL]

kable(tab)
```

var	Pro-Soviet	Turnout
All monuments removed	1.5^{**}	1.5^{**}
	(0.5)	(0.4)
Number of removed monuments	1.4^{**}	1.4^{***}
	(0.4)	(0.3)
One monument removed	1.6^{**}	1.7^{***}
	(0.5)	(0.4)
Two monuments removed	1.9	1.5
	(0.9)	(0.7)
Percentage of monuments removed	1.6^{**}	1.7^{***}
	(0.5)	(0.4)

```
#################################################################
# Table B.6. Appendix B.6 (Floor effects)
#################################################################
q <- D[post==1, quantile(ru_turnout, c(.25, .5))]
out_robust <- list(
  outcomes(D = D, m = models[[3]]),
  outcomes(D = D[ name %in% D[post==1 & ru_turnout >= q[1], name]], m = models[[3]]),
  outcomes(D = D[ name %in% D[post==1 & ru_turnout >= q[2], name]], m = models[[3]])
)

tab <- lapply(1, function(i) lapply(out_robust, function(z) coef_out(z[[i]], name = "X", below = FALSE,
n <- lapply(out_robust, function(x) x[[1]]$N/2)%>%unlist()%>%prettyNum(big.mark = ",")
e <- c("All precincts", paste("Pro-Soviet turnout above", roundr(q, 1), "\\% (", c("25th", "50th"), "per

data.table(e, tab, n) %>% kable()
```

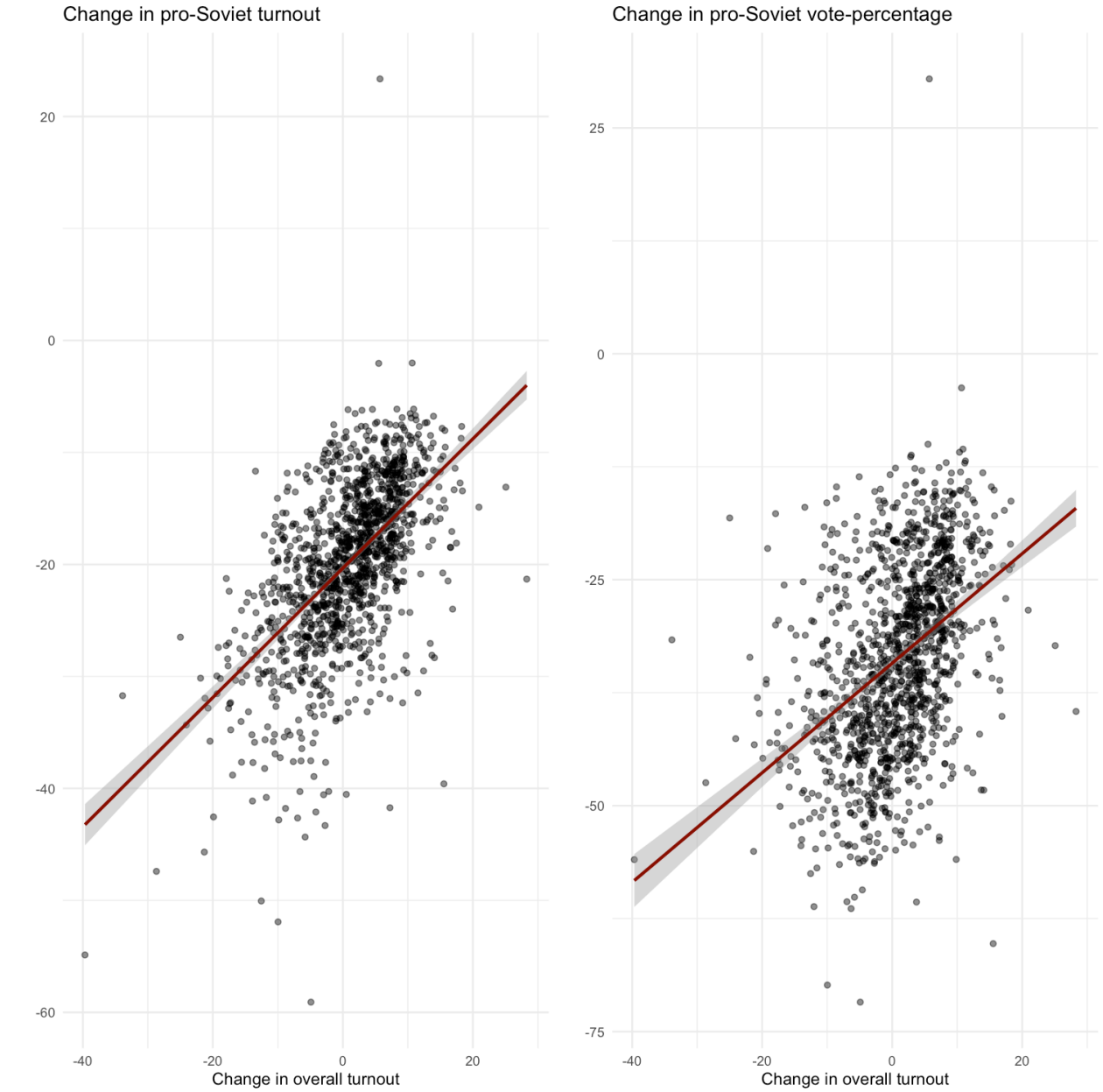
e	b	n
All precincts	$1.6~(0.5)^{**}$	1,296
Pro-Soviet turnout above 2.8 \% (25th percentile)	$1.6~(0.6)^{**}$	973
Pro-Soviet turnout above 6.1 \% (50th percentile)	$1.5~(0.7)$	648

```
#####################################################################
# Figure 2. Appendix B.5: Sensitivity to exclusion of oblasts
#####################################################################
D <- D_data(c("2012-10-28", "2014-05-25"))
oblast_plot <- lapply(unique(D$oblast), function(x) outcomes(D[!oblast %in% x], models[[3]]) %>% lapply(
oblast_plot[, x := stri_trans_general(x, "ukrainian-latin/BGN") %>% gsub("oblast", "", .) %>% str_trim()

ggplot(data = oblast_plot, aes(color = V2, x = factor(x)), color = c("black", "grey")) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_point(aes(y = b), size = 3, position = position_dodge(width = 1/2)) +
  geom_errorbar(aes(ymin = l, ymax = u), width=.2, size = 2, position = position_dodge(width = 1/2)) +
  theme_bw() + ylab("Diff-in-diff coefficient") + xlab("") +
  scale_color_manual(values=c("dark grey", "black")) +
  coord_flip() + theme(
    legend.position="top",
    legend.title = element_blank(),
    legend.text=element_text(size = 14),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.text=element_text(size = 14),
    axis.text.x=element_text(size = 12, face = "bold", color = "black"),
    axis.title=element_text(size = 12))
```




```
#####################################################################
# Figure 3. Appendix C:
#####################################################################

p_load(gridExtra)
D[, ru_share := 100*(pro_russian/votes)]
grid.arrange(
  ggplot(data = D[, list(turnout = turnout[time == 7] - turnout[time == 6], ru_turnout = ru_turnout[time
    geom_point(alpha = 0.5) + geom_smooth(col = "dark red", method = "lm") +
    ylab("") + xlab("Change in overall turnout") +
    ggtitle("Change in pro-Soviet turnout")  + theme_minimal(),
  ggplot(data = D[, list(turnout = turnout[time == 7] - turnout[time == 6], ru_share = ru_share[time ==
    geom_point(alpha = 0.5) + geom_smooth(col = "dark red", method = "lm") +
    xlab("Change in overall turnout") + ylab("") +
```

```
    ggtitle("Change in pro-Soviet vote-percentage") + theme_minimal(), ncol  = 2)

## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



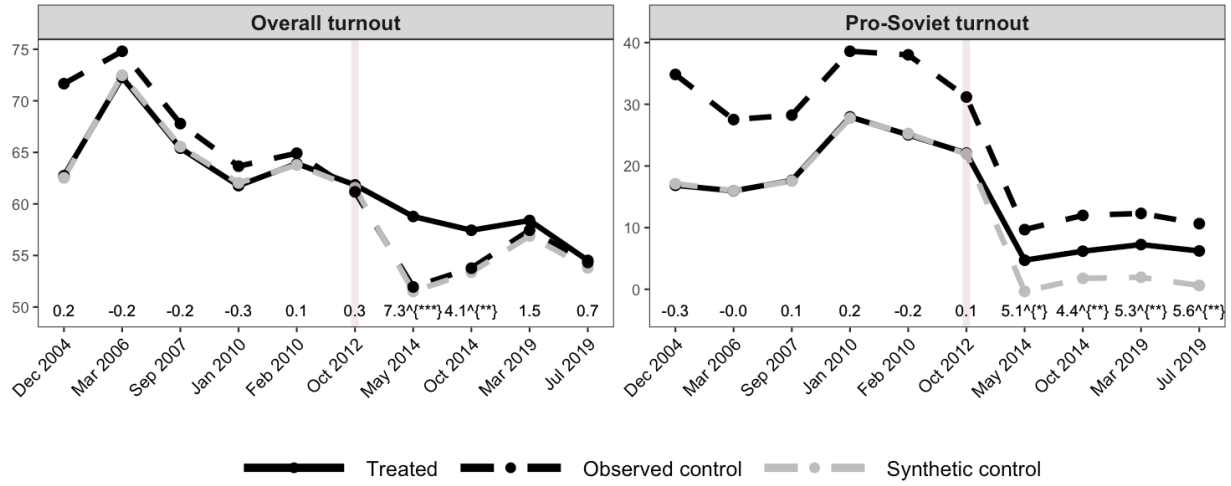

```
#################################################################
# Table C.8. Appendix C.2: Effects on other parties
#################################################################
nat_out <- list(
  felm(update(models[[3]], I(100*nationalist/voters) ~ .), data = D, weights = D$voters),
  felm(update(models[[3]], I(100*(votes - pro_russian - nationalist)/voters) ~ .), data = D, weights = D
  felm(update(models[[3]], I(100*(votes - pro_russian)/voters) ~ .), data = D, weights = D$voters)
)
```

```r
tab <- cbind(
  c("Effect on nationalist turnout",
    "Effect on turnout for parties other than pro-Soviet",
    "Effect on turnout for centrist parties"),
  lapply(nat_out, coef_out, name = "X", below = FALSE, dollar = TRUE) %>% rbindlist())

kable(tab[,-2])
```

V1	b
Effect on nationalist turnout	$-0.5~(0.2)^{*}$
Effect on turnout for parties other than pro-Soviet	$0.4~(0.4)$
Effect on turnout for centrist parties	$-0.1~(0.5)$

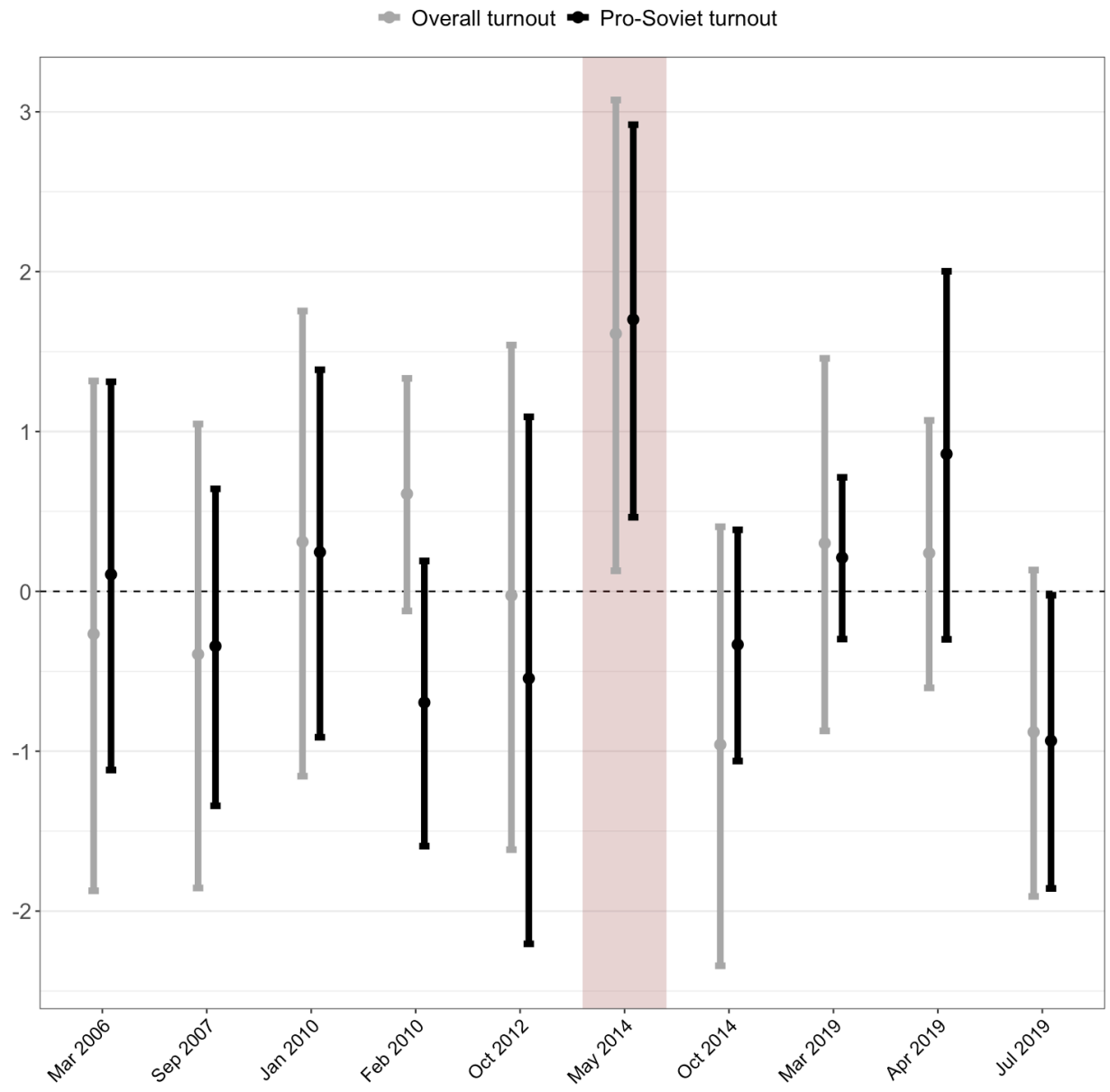
```r
#######################################################
# Figure 3: Generalized synthetic control analysis
#######################################################

# Which units have been treated between 2012 and 2014?
d[, treated := (name %in% d[, (n[election ==  ymd("2014-05-25")] < n[election == ymd("2012-10-28")]), by
# Standard diff-in-diff treatment indicator:
d[ , D := 1*(treated == TRUE & election > ymd("2012-10-28"))]
# Indicator for presidential election
d[, presidential := election %in% ymd("2004-12-25", "2010-01-17", "2010-02-07", "2014-05-25", "2019-03-3
# Rescale turnout to remove the saw-chain pattern
d[, turnout_m := mean(turnout), by = presidential][, turnout_m := turnout - turnout_m + mean(turnout)]

# Fit gsynth
m <- list("turnout_m", "ru_turnout")
g_out <- lapply(m, function(m) quiet(gsynth(Y = m, D = "D", weight = "voters", index = c("name","cycle")

## Bootstrap cluster-corrected p-values
boot_p <- function(out, M){
  g <- function(out){
    # Data
    D <- d[!name %in% out$tr.remove.id, ]
    # Sample regions
    r <- D[, list(oblast = sample(unique(oblast), replace = TRUE))]
    # Track bootstraps
    r[, oblast_boot := .I]
    # Sample units within each sampled region
    f <- function(r) D[oblast == r, sample(unique(name), replace = TRUE)]
    # Replications per unit
    r <- r[, list(name = f(oblast)), by = .(oblast, oblast_boot)][, .(name)][, fake_name := paste(name,
    # Boostrapped data
    D <- merge(r, D, by = "name", allow.cartesian = TRUE)
    # gsynth estimation on bootstrapped data
    gsynth(Y = out$Y, D = "D", weight = "voters", index = c("fake_name","cycle"), data = D, force = "two
  }
  t(replicate(M, g(out)))
}

invisible(quiet(b <- lapply(g_out, boot_p, M = 100)))

# Boostrapped p-values based on T(G-1) distribution with finite
```

```
# sample correction following Cameron / Gehlbach / Miller (2008)
Z  <- d[!name %in% g_out[[1]]$tr.remove.id, ]
G  <- length(unique(Z$oblast))
# Paremeters = # of precincts + # time effects + # coefficients
K  <- length(unique(Z$name)) + 2*length(unique(Z$cycle))
N  <- nrow(Z)
# Adjust S.E. for the degrees of freedom (Cameron/Gehlbach/Miller)
se <- lapply(b, function(x) apply(x, 2, sd)*sqrt(G/(G-1)*(N-1)/(N-K)))
p <- lapply(1:2, function(i) 2*pt(abs(g_out[[i]]$att/se[[i]]), df = G - 1, lower.tail = FALSE))
# Bonferroni adjustment
p <- lapply(p, p.adjust, method = "bonferroni")

g_plot <- function(i) data.table(outcome = m[[i]],
                                 x = 1:length(unique(d$election)),
                                 g_out[[i]]$Y.bar,
                                 ATT = g_out[[i]]$att,
                                 p = p[[i]]
)
d_out <- lapply(1:2, g_plot) %>% rbindlist() %>% melt(measure.vars = grep("Y\\.", colnames(.), value = T

d_out[, att_lab := ""]
d_out[, att_lab := cutoff(ATT, p)]
d_out[, outcome := ifelse(grepl("ru", outcome), "Pro-Soviet turnout", "Overall turnout") %>% as.factor()
d_out[variable == "Y.tr.bar", group := "Treated"]
d_out[variable == "Y.ct.bar", group := "Synthetic control"]
d_out[variable == "Y.co.bar", group := "Observed control"]
d_out[, group := relevel(factor(group), ref = "Treated")]

ggplot(d_out, aes(x = x, y = value, linetype = group, color = group, size = group)) +
  geom_line(size = 1.5) + geom_point(size = 2.5) +
  scale_x_continuous(breaks = 1:length(unique(d$election)), labels = unique(d$election) %>% ymd() %>% so
  facet_wrap(. ~ relevel(outcome, ref = "Overall turnout"), scales = "free_y") + theme_bw() + theme(
  aspect.ratio = 1/2,
  axis.text.x = element_text(size=10, angle=45, vjust = 1, hjust = 1, color = "black"),
  panel.grid.minor = element_blank(),
  panel.grid.major = element_blank(),
  legend.position="bottom",
  legend.text = element_text(size = 12, colour = "black"),
  legend.title = element_blank(),
  strip.text.x = element_text(size = 12, face = "bold"),
  legend.key.width = unit(5, "lines")
) + xlab("") + ylab("") +
  geom_vline(xintercept = which(unique(d$election) %>% sort() == "2012-10-28"), col = "dark red", alpha
  geom_text(aes(label = att_lab, y = -Inf), vjust = -1, size = 3.2) + scale_y_continuous(expand = expans
  scale_linetype_manual(values = c(1, 2, 2)) +
  scale_color_manual(values = c("black", "black", "grey")) +
  scale_size_manual(values = c(1.5, 1, 1)) +
  guides(linetype = guide_legend(override.aes = list(size = 2.25)),
         color = guide_legend())
```



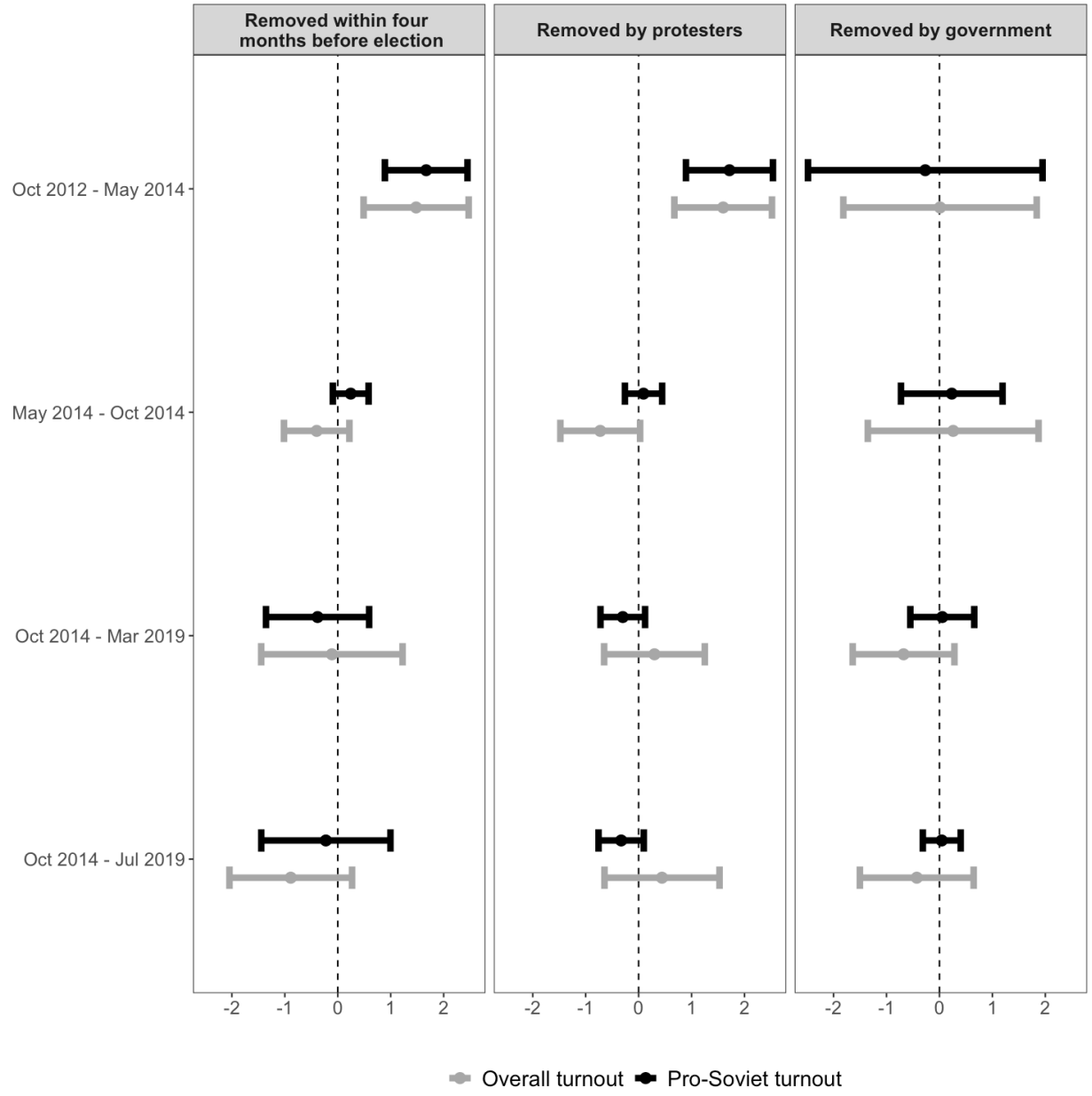

```
########################################
# Figure 4: A falsification test
########################################
# Clean up (load the dataset anew)
d <- readRDS("main_data.RDS")

D_f <- function(d, elections){
  # Select precincts that have any statues at the start of the "cycle":
  D <- d[name %in% d[n > 0 & election %in% ymd(elections), name], ]
  # Indicator for treated unit
  D[, treated := (n[election == elections[1]] > n[election == elections[2]]), by = name]
  D[, cycle := as.numeric(factor(election))]
  D[, (c("oblast", "election", "name", "size")) := lapply(.SD, as.factor), .SDcols = c("oblast", "electi
  D[]
```

```
}

e <- d[, unique(election) %>% ymd() %>% sort()]

out_b <- function(model){
  D <- D_f(d, elections = c("2012-10-28", "2014-05-25"))
  x <- rep(1:length(e), each = 2)
  x <- matrix(x[-c(1, length(x))], nrow = 2)
  lapply(1:ncol(x), function(z){
    dd <- D[cycle %in% x[,z]][, post := 1*(cycle == max(cycle))][, X := post*treated][, t:=cycle]
    outcomes(D = dd, m = model) %>% lapply(function(b) data.table(outcome = b$lhs, election = e[x[2, z]]
  }) %>% rbindlist()
}

b <- out_b(model = models[[3]])

ggplot(data = b, aes(color = factor(outcome, levels = c("turnout", "ru_turnout")), x = factor(election)
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_vline(xintercept = which(e[-1]=="2014-05-25"), lwd = 25, color = "dark red", alpha = 0.2) +
  geom_point(aes(y = b), size = 3, position = position_dodge(width = 1/3)) +
  geom_errorbar(aes(ymin = l, ymax = u), width = .2, size = 2, position = position_dodge(width = 1/3)) +
  theme_bw() + xlab("") + ylab("") +
  scale_x_discrete(labels = as.yearmon(e[-1])) +
  scale_color_manual(labels = c("Overall turnout", "Pro-Soviet turnout"), values=c("dark grey", "black")
  theme(
    legend.position="top",
    legend.title = element_blank(),
    legend.text=element_text(size = 14),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    axis.text=element_text(size = 14),
    axis.text.x=element_text(size = 12, color = "black", angle = 45, hjust = 1),
    axis.title=element_text(size = 12))
```




```
############################################################
#Figure 5: Estimates for different types of removals
############################################################

# Create two-period data
D_het <- function(elections){
  setDT(lenin)
  D <- D_data(elections)
  # Check the type of removal by identifying which statue has disappeared:
  which_ids <- function(x1, x2) setdiff(strsplit(x1, ";") %>% unlist(), strsplit(x2, ";") %>% unlist())
  D[removed_any == 1, removed_by_activists := any(grepl("activists", lenin[id %in% which_ids(statue_id[1
  D[removed_any == 0, removed_by_activists := FALSE]
  # Count only removals within X months prior to a given election:
  D[removed_any == 1, removed_days_before := (max(ymd(election))  - ymd(max(lenin[id %in% which_ids(stat
```

```
  D[is.na(removed_days_before), removed_days_before := 0]
  D
}

out_each <- list(
  lapply(x, function(x) outcomes(D_het(x)[, X := 1*(removed_any == 1 & post == 1 & removed_days_before <
  lapply(x, function(x) outcomes(D_het(x)[, X := 1*(removed_any == 1 & post == 1 & removed_by_activists
  lapply(x, function(x) outcomes(D_het(x)[, X := 1*(removed_any == 1 & post == 1 & removed_by_activists
)

# Which elections are being compared?
els <- lapply(x, function(z) paste(as.yearmon(z), collapse = " - ")) %>% unlist()
# How treatment is defined?
removed_by <- c("Removed within four \n months before election",
                "Removed by protesters",
                "Removed by government")

cofs <- lapply(1:3, function(i) data.table(lapply(out_each[[i]], function(z) lapply(z, function(z) data.

ggplot(data = cofs, aes(color = y, x = factor(e, rev(els)))) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_point(aes(y = b), size = 3, position = position_dodge(width = 1/3)) +
  geom_errorbar(aes(ymin = l, ymax = u), width=.2, size = 2, position = position_dodge(width = 1/3)) +
  theme_bw() + ylab("") + xlab("") +
  scale_color_manual(values=c("dark grey", "black")) +
  facet_grid(. ~ factor(removed_by, levels = unique(cofs$removed_by))) +
  coord_flip() +
  theme(
  legend.position="bottom",
  legend.title = element_blank(),
  legend.text=element_text(size = 14),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  axis.text=element_text(size = 12),
  axis.text.x=element_text(size = 12),
  axis.title=element_text(size = 12),
  strip.text = element_text(size=12, face = "bold")
)
```

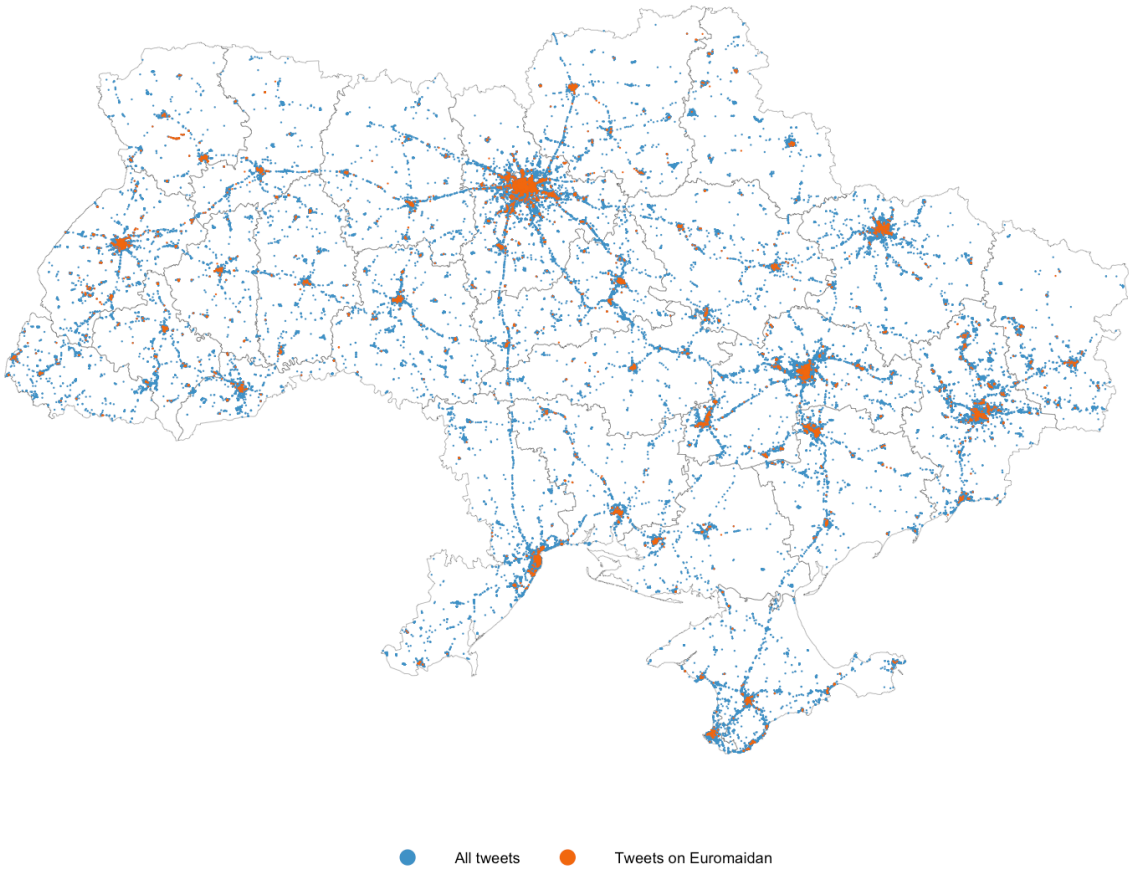



```
#########################################################################
# The Role of Protests
#########################################################################

# Load UIK polygons from 2019 and use as the main template
uik <- readRDS("uik_polygons.RDS")

#########################################################################
# Figure 4. Appendix D.1:
#########################################################################

# Protest data from media (expand to by-day format)
protest <- readRDS("protests.RDS")
ggplot() + geom_sf(data = ukr, fill = "white", lwd = 0.1, alpha = 0.1) +
```

```
  coord_sf(crs = st_crs(ukr), datum = NA) +
  geom_sf(data = protest, alpha = .75, pch = 16, color = "dark red") +
  coord_sf(crs = st_crs(protest), datum = NA) +
  theme_minimal()

## Coordinate system already present.  Adding new coordinate system, which will replace the
existing one.
```

```
# Merge protest data with other geographies
protest <- st_join(uik[, c("name", "ADM2_PCODE", "ADM3_PCODE")], protest) %>% st_set_geometry(NULL) %>%

# Load twitter data
twitter <- readRDS("twitter.RDS")
# Map on the uik template (merge by unique coordinate and then map back, to save computation time)
```

```
D <- twitter[, .(lon, lat)] %>% unique() %>% st_as_sf(coords = c("lon", "lat"), crs = st_crs(uik), remov
D <- st_join(D, uik[, c("name", "ADM2_PCODE", "ADM3_PCODE")]) %>% st_set_geometry(NULL)
# Map back the merged attributes on the main twitter frame
twitter <- twitter[D, on = c("lon", "lat")]

########################################
# Figure 5. Appendix D.2:
########################################
D <- twitter[, .N, by = .(lon, lat, includes_maidan)]
D <- st_as_sf(D, coords = c("lon", "lat"), crs = st_crs(ukr))
cols <- c(brewer.pal(9, "Blues")[6], brewer.pal(9, "Oranges")[6])
ggplot() + geom_sf(data = ukr, fill = "white", lwd = 0.1, alpha = 0.1) +
  coord_sf(crs = st_crs(ukr), datum = NA) +
  geom_sf(data = D, aes(color = factor(includes_maidan)), alpha = 1, cex = 0.2, pch = 16) +
  coord_sf(crs = st_crs(D), datum = NA) +
  theme_minimal() +
  scale_color_manual(name = "", values = cols, labels = c("All tweets", "Tweets on Euromaidan")) +
  theme(legend.position="bottom", legend.key.size = unit(3,"line")) +
  guides(color = guide_legend(override.aes = list(size = 4)))

## Coordinate system already present.  Adding new coordinate system, which will replace the
existing one.
```

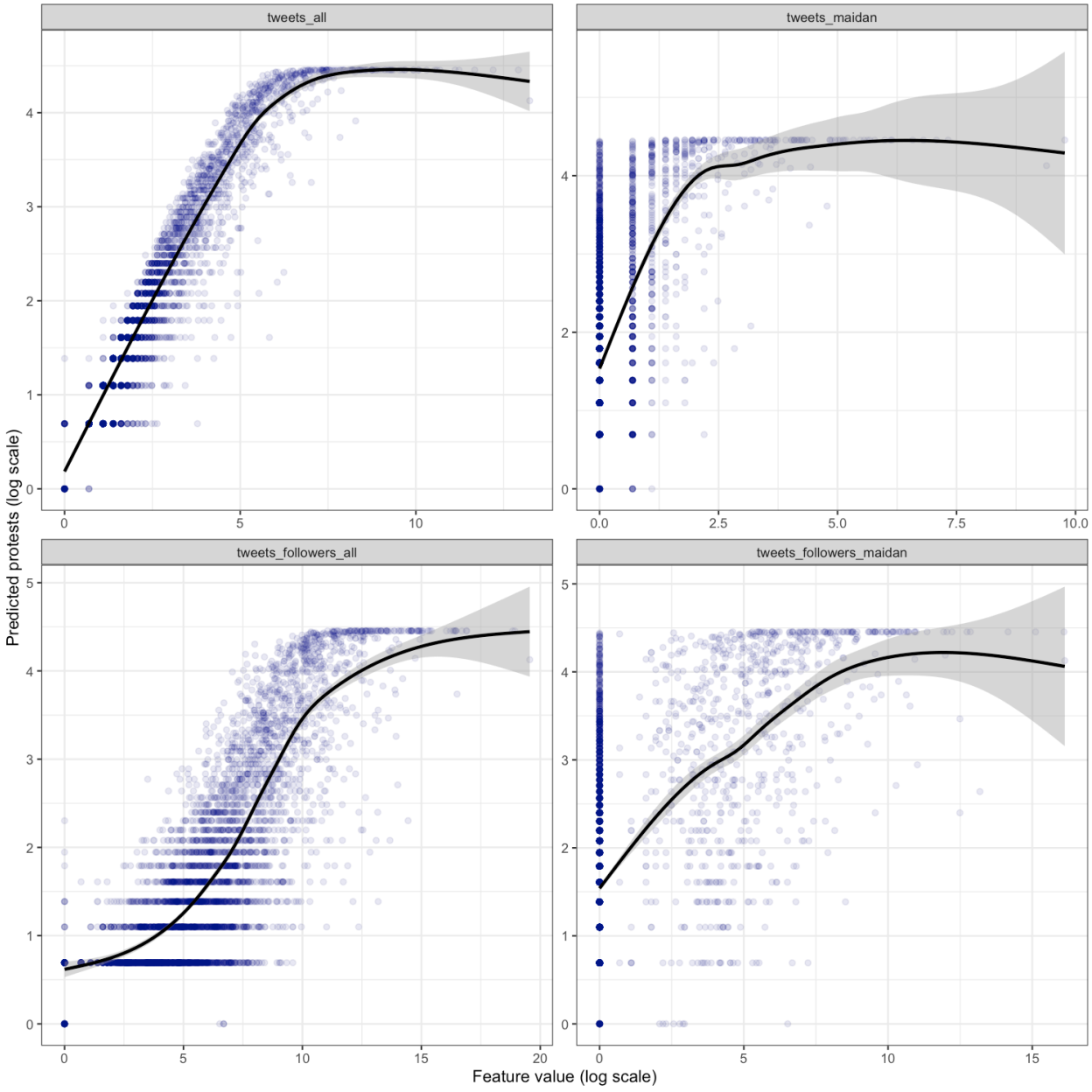



```
# This function that merges datasets and produces DiD estimates at the level given by "unit" value; "va

z_out <- function(unit = "ADM3_PCODE", variables = NULL){

  # Aggregating protests over the revelant units
  p <- protest[ , c(unit, "start_date", "end_date"), with = FALSE]
  p <- p[!is.na(start_date), list(date = seq(mdy(start_date)[1], mdy(end_date)[1], by = 1)), by = unit]

  # Prepare election data for cycles 5:6
  d <- readRDS("main_data.RDS")
  d <- d[election %in% ymd(c("2012-10-28", "2014-05-25"))]
  # Aggregate election/monument data to unit level
  d[, unit := get(unit)]
  d <- d[, list(
```

```
  voters = sum(voters),
  votes = sum(votes),
  turnout = 100*sum(votes)/sum(voters),
  ru_turnout = 100*sum(pro_russian)/sum(voters),
  statues = sum(n),
  dist_to_kiev = mean(dist_to_kiev),
  size_small = mean(size == "mala"),
  size_large = mean(size == "velyka"),
  lon = mean(lon),
  lat = mean(lat),
  km_road = mean(km_road),
  post = 1*(election == "2014-05-25")
), by = .(ADM1_UA, unit, election)]
d[, statue_exists := 1*(unit %in% d[statues > 0 & election == "2012-10-28", unit])]
d[, statue_removed := 1*(statues[1] > statues[2]), by = unit]

##############################################
# Merge & analyze (predict protests by day)
##############################################

# Twitter data by unit-date
D <- twitter[!is.na(get(unit)), list(
  tweets = .N,
  tweets_followers = sum(followers_count)), by = c(unit, "includes_maidan", "date")]
D[ , maidan := ifelse(includes_maidan == 0, "all", "maidan")]
D <- dcast(D, get(unit) + date ~ maidan, value.var = c("tweets", "tweets_followers"), fill = 0)

# Create unit-day template and merge twitter and protest data on the top of it
data <- CJ(date = seq(ymd("2013-12-01"), ymd("2014-02-24"), by = 1), unit = D[, unit] %>% unique())
data <- merge(data, D, by = c("unit", "date"), all.x = TRUE)
data[p, on = c("unit", "date"), protest := i.protest]
cols <- c("tweets_all", "tweets_maidan", "tweets_followers_all", "tweets_followers_maidan", "protest")
data[, (cols) := lapply(.SD, function(x) replace_na(x, replace = 0)), .SDcols = cols]

cols <- grep("tweets", colnames(data), value = TRUE)

# Design matrix for predictions
cat("Fitting random forest", '\n')
rf_out <- randomForest(factor(protest) ~ tweets_all + tweets_maidan + tweets_followers_all +
                         tweets_followers_maidan, data = data,
                replace = TRUE,
                strata = data$protest,
                sampsize = c(sum(data$protest), sum(data$protest)),
                ntree = 1000)
data[, protest_predict := predict(rf_out, type = "prob")[,2]]
cat("Area under ROC = ", roc.area(data$protest, data$protest_predict)$A, "\n")
cat("Accuracy = ", (data[protest==1, sum(protest_predict >= 1/2)] + data[protest==0, sum(protest_predi

p_data <- data[, list(
  protests_predicted = log(1 + sum(protest_predict > 1/2)),
  tweets_all = log(1 + sum(tweets_all)),
  tweets_maidan = log(1 + sum(tweets_maidan)),
  tweets_followers_all = log(1 + sum(tweets_followers_all)),
```

```
    tweets_followers_maidan = log(1 + sum(tweets_followers_maidan))
  ), by = unit] %>% melt(id.vars = c("unit", "protests_predicted"))

  # Figure 6. Appendix D.2:
  print(ggplot(p_data, aes(x = value, y = protests_predicted)) +
          geom_point(alpha = 0.1, col = "dark blue") +
          geom_smooth(color = "black") +
          facet_wrap(. ~ variable, scales = "free") +
          theme_bw() + ylab("Predicted protests (log scale)") +
          xlab("Feature value (log scale)"))

  # Now map predictions to the the complete dataset of unit-by-day:
  D <- merge(CJ(date = unique(data$date), unit = unique(d[, unit])), data, by = c("unit", "date"), all.x
  cols <- grep("tweets|protest", colnames(D), value = TRUE)
  D[, (cols) := lapply(.SD, function(x) replace_na(x, replace = 0)), .SDcols = cols]
  X <- D[,grep("tweets", colnames(data), value = TRUE), with = FALSE]
  D[, protest_predict := predict(rf_out, newdata = D, type = "prob")[,2]]

  # Aggregate twitter and protest data to unit level and merge to election/monument data
  data[, protest_cut := protest_predict > 0.5]
  data <- merge(d, D[, lapply(.SD, sum),
                     by = unit,
                     .SDcols = grep("protest|tweets", names(D), value = TRUE)],
                by = "unit", all.x = TRUE)
  cat("Units with none observed = ", data[election == "2012-10-28"][protest == 0, .N], "\n")
  cat("Predict protests where none observed = ", data[election == "2012-10-28"][protest == 0, sum(protes

  # Use only councils with two time periods (removes one observation):
  data <- data[unit %in% data[, .N, by = unit][N==2, unit]]

  # Regressions
  m_out <- list()

  for (i in variables){
    data[, Z := get(i)]
    m <- Formula(y ~ post:(statue_removed  + log(1 + Z) + log(dist_to_kiev) +
                             lon*lat + size_small + size_large +
                             log(km_road)) | unit + factor(election)*factor(ADM1_UA) | 0 | unit + ADM1_U
    m_out[[i]] <-  list(
      outcomes(D = data[statue_exists == 1], m = m),
      outcomes(D = data[statue_exists == 0], m = update(m, . ~ . - post:statue_removed)))
  }

  attr(m_out, "unit") <- unit
  attr(m_out, "variables") <- variables

  m_out

}

################################################
# Protest/twitter
################################################
```

```r
tab_out <- function(z, Unit = ""){
  unit <- attr(z, "unit")
  variables <- attr(z, "variables")
  tab <- lapply(1:3, function(k){
    lapply(1:2, function(i){lapply(z[[k]][[i]], function(x){
      b <- coef_out(x, name = "removed|Z|tweets", dollar = FALSE, below = TRUE)
      data.table(outcome = attributes(x$response)[[2]][[2]], b, N = prettyNum(x$N, big.mark=",",scientif
        rbindlist()}) %>% rbindlist() %>% data.table(., Z = variables[k])}) %>% rbindlist()
  tab[, outcome := ifelse(grepl("ru_turnout", outcome), "Pro-Soviet turnout", "Overall turnout") %>% fac
  tab[, var := ifelse(grepl("Z", var), "Protests", "Removals")]
  tab[, Z:=factor(Z, levels = variables)]
  tab <- dcast(tab, Z + var + type ~ subset + outcome, value.var = c("b"))
  tab[type == "SE", var := ""]
  n <- lapply(1:2, function(x) z[[1]][[x]][[1]]$N) %>% unlist() %>% prettyNum(big.mark=",",scientific=FA
  tab[,-c(1,3)]
}

vars <- c("protest", "tweets_followers_maidan", "protest_predict")

############################################################
# Table 4: Protest as an alternative mechanism
############################################################
z <- z_out(unit = "ADM3_PCODE", variables = vars) %>% tab_out(Unit = "Councils")

## Fitting random forest
## Area under ROC =  0.8316397
## Accuracy =  0.8559411

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



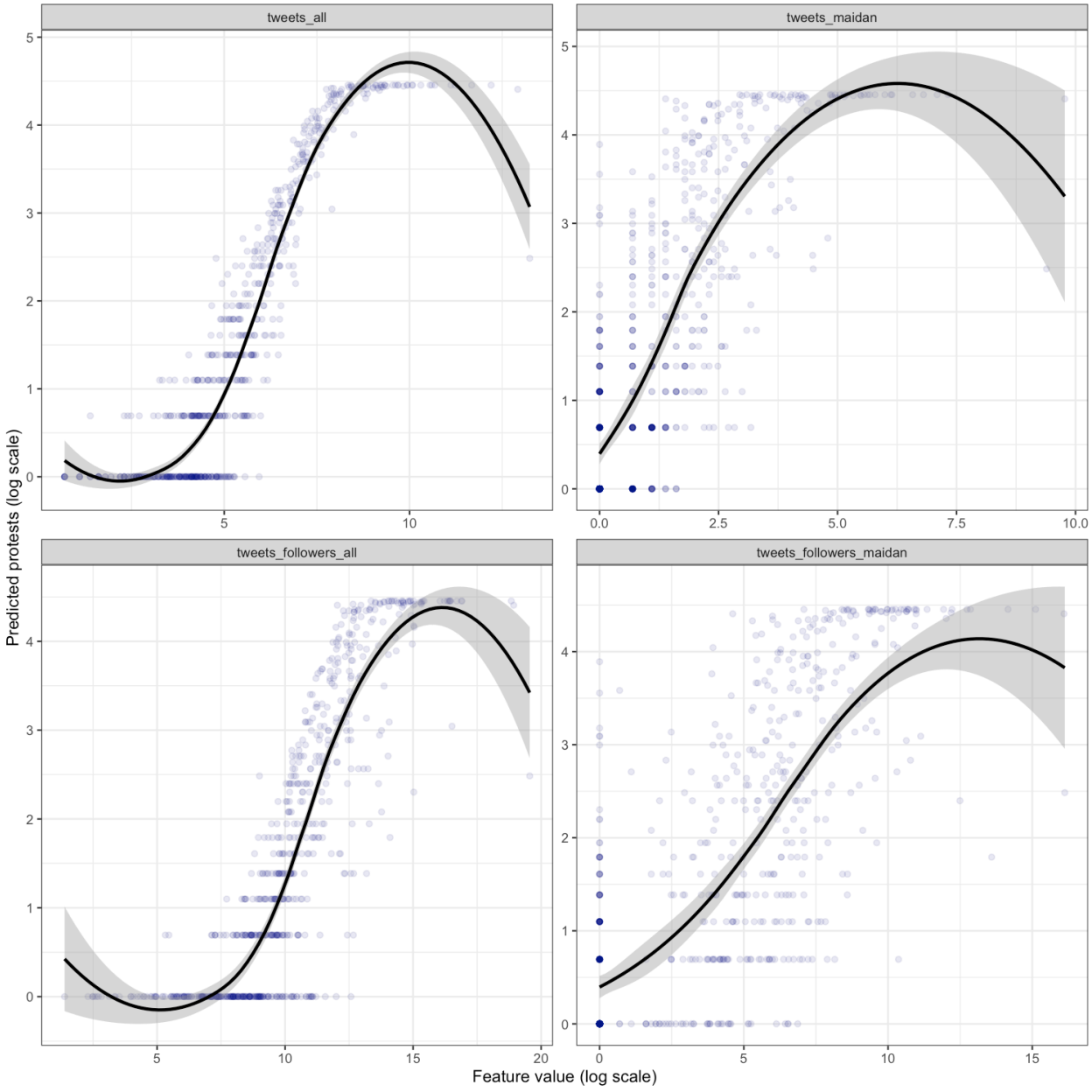

```
## Units with none observed =  9134
## Predict protests where none observed =  3462

kable(z[,1:3])
```

var	With monuments_Overall turnout	With monuments_Pro-Soviet turnout
Protests	0.4^{***}	0.4^{*}
	(0.1)	(0.2)
Removals	2.6^{***}	2.4^{***}
	(0.5)	(0.6)
Protests	0.3^{***}	0.2^{**}
	(0.0)	(0.1)
Removals	2.4^{**}	2.5^{**}
	(0.7)	(0.8)
Protests	0.2	0.2
	(0.2)	(0.2)
Removals	3.3^{***}	3.1^{***}
	(0.6)	(0.7)

```
kable(z[,c(1, 4:5)])
```

var	Without monuments_Overall turnout	Without monuments_Pro-Soviet turnout
Protests	-0.0	-0.0
	(0.2)	(0.3)
Removals	NA	NA
	NA	NA
Protests	0.1	0.0
	(0.1)	(0.1)
Removals	NA	NA
	NA	NA
Protests	0.6	0.8^{**}
	(0.3)	(0.3)
Removals	NA	NA
	NA	NA

```
############################################################
# Table D.9. Appendix D.3:
############################################################
z <- z_out(unit = "ADM2_PCODE", variables = vars) %>% tab_out(Unit = "Rayons")

## Fitting random forest
## Area under ROC =  0.7337558
## Accuracy =  0.8159953

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
## Units with none observed =  458
## Predict protests where none observed =  453

kable(z[,1:3])
```

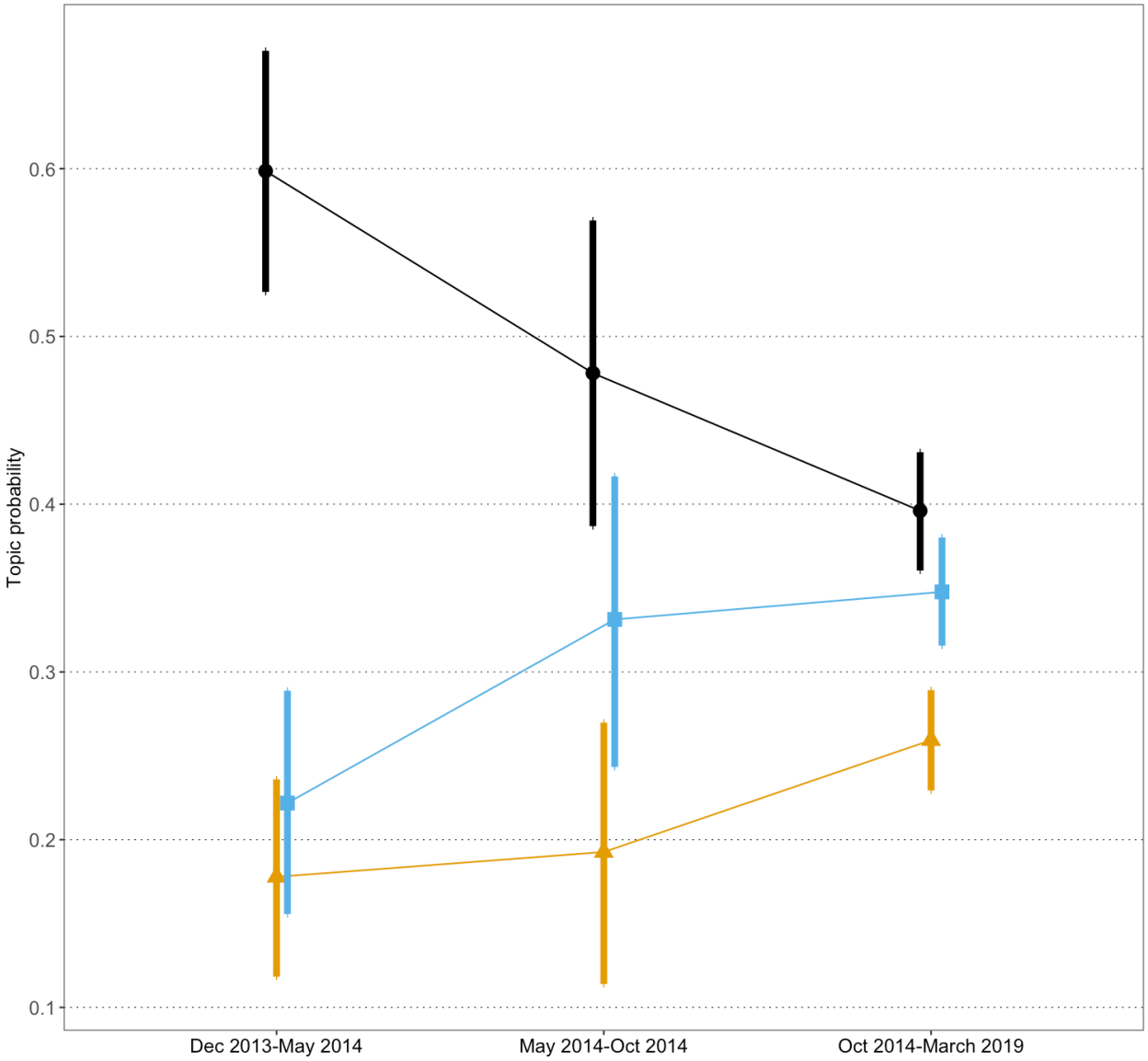
var	With monuments_Overall turnout	With monuments_Pro-Soviet turnout
Protests	0.4^{*}	0.3
	(0.1)	(0.2)
Removals	2.3^{***}	2.3^{***}
	(0.5)	(0.4)
Protests	0.3^{***}	0.3^{***}
	(0.0)	(0.0)
Removals	2.3^{***}	2.3^{***}
	(0.5)	(0.4)
Protests	0.6^{***}	0.9^{***}
	(0.1)	(0.2)
Removals	2.5^{***}	2.5^{***}
	(0.5)	(0.4)

```
kable(z[,c(1, 4:5)])
```

var	Without monuments_Overall turnout	Without monuments_Pro-Soviet turnout
Protests	0.1	0.3
	(0.1)	(0.1)
Removals	NA	NA
	NA	NA
Protests	0.0	-0.1
	(0.1)	(0.2)
Removals	NA	NA
	NA	NA
Protests	0.4	0.2
	(0.4)	(0.5)
Removals	NA	NA
	NA	NA

```
############################################################
# Structural topic analysis
############################################################
l <- readRDS("stm_text.RDS")
processed <- textProcessor(l$text, metadata = data.frame(l))

## Building corpus...
## Converting to Lower Case...
## Removing punctuation...
## Removing stopwords...
## Removing numbers...
## Stemming...
## Creating Output...

out <- prepDocuments(processed$documents, processed$vocab, processed$meta)

## Removing 14240 of 29995 terms (14240 of 225227 tokens) due to frequency
## Your corpus now has 771 documents, 15755 terms and 210987 tokens.

st_out <- stm(documents = out$documents, vocab = out$vocab, K = 3, prevalence = ~ factor(period), max.em

simBetas <- function (parameters, nsims = 1000){
  simbetas <- list()
  for (i in 1:length(parameters)) {
    simbetas[[i]] <- do.call(rbind, lapply(parameters[[i]], function(x) rmnorm(n = nsims, mean = x$est,
```

```
  }
  apply(simbetas[[1]], 2, quantile, c(.025, .5, .975)) %>% t()
}

# Topic words
words <- apply(labelTopics(st_out)$prob, 1, paste, collapse = ", ")
en_words <- c("Russia, country, war, Soviet, Russian, power, state",
              "conversation, first, people, new, film, book",
              "monument, city, Donetsk, places, local, Kyiv, people")
words <- paste(c("{\\bf Sovereignty}", "{\\bf Culture}", "{\\bf Protest}"), en_words, sep = ": ")

p_data <- lapply(1:3, function(i){
  i <<- i
  st_reg <- estimateEffect(c(i) ~ -1 + factor(period), st_out,  meta = out$meta, uncertainty = "Global")
  data.table(period = 1:3, topic = words[i], simBetas(parameters = st_reg$parameters))
}) %>% rbindlist() %>% setnames(c("period", "topic", "l", "b", "u"))
p_data$topic <- factor(p_data$topic, levels = words[3:1])

#####################################
#Figure 6: The meaning of Lenin
#####################################

ggplot(p_data, aes(x = period, y = b, color = topic, shape = topic)) +
  geom_point(position = position_dodge(width = 1/10), size = 4) +
  geom_line(position = position_dodge(width = 1/10)) +
  geom_errorbar(aes(ymin = l, ymax = u), width=.01, size = 2, position = position_dodge(width = 1/10)) +
  scale_x_discrete(limits = c(1:3), labels = c("Dec 2013-May 2014", "May 2014-Oct 2014", "Oct 2014-March
  xlab("") + ylab("Topic probability") +
  theme_bw() +
  scale_color_manual(values=c("#000000", "#E69F00", "#56B4E9")) +
  theme(
  legend.position="bottom",
  legend.title = element_text(size = 12),
  legend.text=element_text(size = 12),
  panel.grid.minor = element_blank(),
  panel.grid.major.x = element_blank(),
  panel.grid.major.y = element_line(linetype = "dotted", color = "black", size = .25),
  axis.text=element_text(size = 12),
  axis.text.x=element_text(size = 12, color = "black", angle = 0, hjust = .5),
  axis.title=element_text(size = 12))

## Warning:  Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```



, city, Donetsk, places, local, Kyiv, people {\bf Culture}: conversation, first, people, new, film, book {\bf Sovereignty}: Russia, cc

```
# This is to extract the point estimates reported in the text (not tables or figues)
est_b <- estimateEffect(1:3 ~ factor(period, levels = c(2, 1, 3)), st_out,  meta = out$meta, uncertainty
est_b$tables

## [[1]]
##                                       Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)                           0.33355782 0.04368491  7.6355383 6.686990e-14
## factor(period, levels = c(2, 1, 3))1 -0.11157745 0.05555292 -2.0084895 4.494092e-02
## factor(period, levels = c(2, 1, 3))3  0.01312913 0.04643674  0.2827316 7.774588e-01
##
## [[2]]
##                                       Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)                           0.19147309 0.03901125  4.9081503 1.123228e-06
## factor(period, levels = c(2, 1, 3))1 -0.01691412 0.04941033 -0.3423195 7.322041e-01
```

```
## factor(period, levels = c(2, 1, 3))3  0.06817340 0.04177337  1.6319824 1.030931e-01
##
## [[3]]
##                                       Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)                          0.47436591 0.04556533 10.410676 7.824688e-24
## factor(period, levels = c(2, 1, 3))1  0.12917186 0.05854427  2.206396 2.765152e-02
## factor(period, levels = c(2, 1, 3))3 -0.08068481 0.04849969 -1.663615 9.659733e-02

# How long did it take?
round(Sys.time() - start, 1)

## Time difference of 31.7 mins
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 4.1.1 (2021-08-10)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.7
##
## Matrix products: default
## BLAS:   /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versio
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] grid      datasets  splines   utils     graphics  grDevices stats     methods
## [9] base
##
## other attached packages:
##  [1] gridExtra_2.3       knitr_1.33          stm_1.3.6           tm_0.7-8
##  [5] NLP_0.2-1           verification_1.42   dtw_1.22-3          proxy_0.4-26
##  [9] CircStats_0.2-6     boot_1.3-28         fields_12.5         viridis_0.6.1
## [13] viridisLite_0.4.0   spam_2.7-0          dotCall64_1.0-1     randomForest_4.6-14
## [17] gsynth_1.2.1        zoo_1.8-9           RColorBrewer_1.1-2  ggthemes_4.2.4
## [21] tikzDevice_0.12.3.1 tidyr_1.1.3         lfe_2.8-7           stringr_1.4.0
## [25] stringi_1.7.4       lubridate_1.7.10    dplyr_1.0.7         stringdist_0.9.7
## [29] readxl_1.3.1        sf_1.0-2            sp_1.4-5            jsonlite_1.7.2
## [33] mgcv_1.8-36         nlme_3.1-152        Formula_1.2-4       stargazer_5.2.2
## [37] data.table_1.14.0   xtable_1.8-4        magic_1.5-9         abind_1.4-5
## [41] msm_1.6.8           mnormt_2.0.2        MASS_7.3-54         lme4_1.1-27.1
## [45] Matrix_1.3-4        ggplot2_3.3.5       pacman_0.5.1
##
## loaded via a namespace (and not attached):
##  [1] minqa_1.2.4        colorspace_2.0-2   ellipsis_0.3.2     class_7.3-19
##  [5] rstudioapi_0.13    farver_2.1.0       listenv_0.8.0      SnowballC_0.7.0
##  [9] fansi_0.5.0        mvtnorm_1.1-2      xml2_1.3.2         codetools_0.2-18
## [13] doParallel_1.0.16  nloptr_1.2.2.2     compiler_4.1.1     assertthat_0.2.1
## [17] s2_1.0.6           tools_4.1.1        gtable_0.3.0       glue_1.4.2
## [21] wk_0.5.0           maps_3.3.0         doRNG_1.8.2        tinytex_0.33
## [25] Rcpp_1.0.7         slam_0.1-48        cellranger_1.1.0   vctrs_0.3.8
## [29] filehash_2.4-2     iterators_1.0.13   xfun_0.25          globals_0.14.0
```

```
## [33] lifecycle_1.0.0    rngtools_1.5       future_1.22.1      scales_1.1.1
## [37] parallel_4.1.1     sandwich_3.0-1     expm_0.999-6       highr_0.9
## [41] foreach_1.5.1      e1071_1.7-8        matrixStats_0.60.1 rlang_0.4.11
## [45] pkgconfig_2.0.3    evaluate_0.14      lattice_0.20-44    purrr_0.3.4
## [49] labeling_0.4.2     tidyselect_1.1.1   parallelly_1.27.0  magrittr_2.0.1
## [53] R6_2.5.1           generics_0.1.0     DBI_1.1.1          pillar_1.6.2
## [57] withr_2.4.2        units_0.7-2        survival_3.2-11    tibble_3.1.4
## [61] crayon_1.4.1       KernSmooth_2.23-20 utf8_1.2.2         tmvnsim_1.0-2
## [65] digest_0.6.27      classInt_0.4-3     munsell_0.5.0
```

```
Sys.time()
```

```
## [1] "2021-09-02 12:51:15 EEST"
```